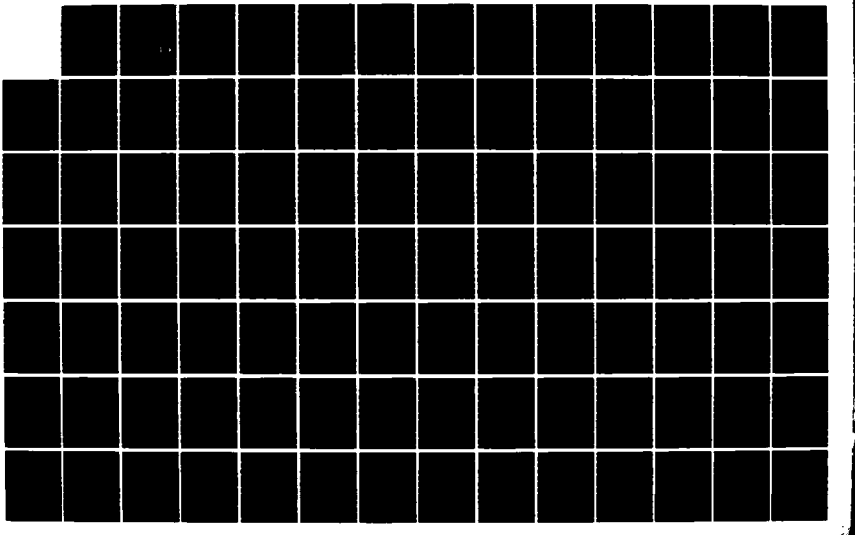
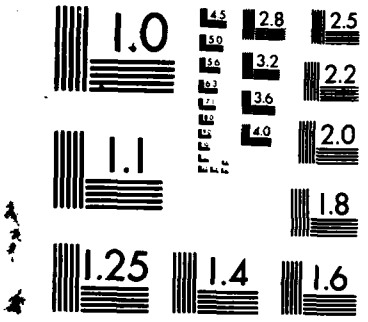
AD-A173 427 COMPUTER PROGRAM DEVELOPMENT SPECIFICATION FOR TACTICAL 1/2
INTERFACE SYSTEM(U) ANALYTICS INC MCLEAN VA 31 JUL 81
1585-TR-02 DAAK80-81-C-0010

UNCLASSIFIED F/G 9/2 NL






MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A173 427

Technical Report 1585-TR-02

# COMPUTER PROGRAM DEVELOPMENT SPECIFICATION FOR TACTICAL INTERFACE SYSTEM (FINAL)

Submitted to:
CENTACS
System Validation Division
Ft. Monmouth, New Jersey 07703

DTIC ELECTE S D OCT 24 1986 A

Contract Number DAAK80-81-C-0010
CDRL Item No. A002

31 July 1981

Prepared by the Staff of Analytics

This document has been approved for public release and sale; its distribution is unlimited.

ANALYTICS
7680 OLD SPRINGHOUSE RD., McLEAN, VA. 22102

86-10-24-045

SDSS-MMP-B1
Code Ident 56496
Part 1 of Two Parts

# PRIME ITEM DEVELOPMENT SPECIFICATION
# FOR
# TACTICAL INTERFACE SYSTEM
## Computer Program, SDSS-MMP-B1

Submitted to:

CENTACS
System Validation Division
Ft. Monmouth, New Jersey 07703

Contract Number DAAK80-81-C-0010
CDRL Item No. A002

31 July 1981

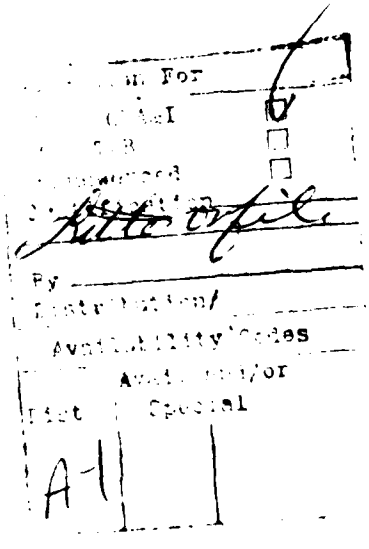


TABLE OF CONTENTS

1. SCOPE . . . . . 1
1.1 Identification . . . . . 1
1.2 Functional summary . . . . . 1

2. APPLICABLE DOCUMENTS. . . . . 2
2.1 Government documents . . . . . 2
2.2 Non-government documents . . . . . 6

3. REQUIREMENTS. . . . . 8
3.1 Computer program definition. . . . . 11
3.1.1 Interface requirements . . . . . 14
3.1.1.1 Interface block diagram. . . . . 14
3.1.1.2 Detailed interface definition. . . . . 14
3.1.1.2.1 PSS interface. . . . . 16
3.1.1.2.1.1 Functional description of PSS interface. . . . . 21
3.1.1.2.1.2 Description of PSS ICE packet interface. . . . . 25
3.1.1.2.1.3 Description of PSS logical I/O interface . . . . . 26
3.1.1.2.1.3.1 PSS initiated write ICE string . . . . . 39
3.1.1.2.1.3.2 TIS initiated write ICE string . . . . . 41
3.1.1.2.1.3.3 Send or receive data . . . . . 42
3.1.1.2.1.3.4 Bootstrap protocol across the SSIM . . . . . 43
3.1.1.2.2 VMS interface. . . . . 43
3.1.1.2.3 RMS interface. . . . . 44
3.1.1.2.4 VAX-11/780 input/output subsystem. . . . . 44
3.1.1.2.5 DR11-B interface . . . . . 49
3.2 Detailed functional requirements . . . . . 55
3.2.1 Interface control element handler (ICEHANDL) . . . . . 69
3.2.1.1 ICEHANDL inputs. . . . . 70
3.2.1.2 ICEHANDL processing. . . . . 71
3.2.1.3 ICEHANDL outputs . . . . . 72
3.2.2 Job control (JOBCNTRL) . . . . . 72
3.2.2.1 JOBCNTRL inputs. . . . . 73
3.2.2.2 JOBCNTRL processing. . . . . 73
3.2.2.3 JOBCNTRL outputs . . . . . 74
3.2.3 Queue operations (QUEUEOPS). . . . . 74
3.2.3.1 QUEUEOPS inputs. . . . . 75
3.2.3.2 QUEUEOPS processing. . . . . 75
3.2.3.3 QUEUEOPS outputs . . . . . 76
3.2.4 File operations (FILEOPS). . . . . 77
3.2.4.1 FILEOPS inputs . . . . . 77
3.2.4.2 FILEOPS processing . . . . . 78
3.2.4.3 FILEOPS outputs. . . . . 79

TABLE OF CONTENTS
(Continued)

3.2.5 Tape controller (TAPECNTRL). . . . . 79
3.2.5.1 TAPECNTRL inputs . . . . . 79
3.2.5.2 TAPECNTRL processing . . . . . 80
3.2.5.3 TAPECNTRL outputs. . . . . 81
3.2.6 AN/GYK-12 downline loader (LOAD) . . . . . 81
3.2.6.1 LOAD inputs. . . . . 83
3.2.6.2 LOAD processing. . . . . 83
3.2.6.3 LOAD outputs . . . . . 85
3.2.7 Data manager (DATAMANGR) . . . . . 86
3.2.7.1 DATAMANGR inputs . . . . . 86
3.2.7.2 DATAMANGR processing . . . . . 86
3.2.7.3 DATAMANGR outputs. . . . . 87
3.2.8 Command interpreter (CMDINTRP) . . . . . 88
3.2.8.1 CMDINTRP inputs. . . . . 88
3.2.8.2 CMDINTRP processing. . . . . 88
3.2.8.3 CMDINTRP outputs . . . . . 89
3.2.9 JCL interpreter (JCLINTRP) . . . . . 89
3.2.9.1 JCLINTRP inputs. . . . . 90
3.2.9.2 JCLINTRP processing. . . . . 90
3.2.9.3 JCLINTRP outputs . . . . . 91
3.2.10 System Initiator/Terminator (SYSINITRM) . . . . . 91
3.2.10.1 SYSINITRM inputs. . . . . 91
3.2.10.2 SYSINITRM processing. . . . . 91
3.2.10.3 SYSINITRM outputs . . . . . 92
3.2.11 User message processor (USERMSGP) . . . . . 93
3.2.11.1 USERMSGP inputs . . . . . 93
3.2.11.2 USERMSGP processing . . . . . 93
3.2.11.3 USERMSGP outputs. . . . . 94
3.2.12 TIS utilities . . . . . 94
3.2.12.1 TIS utilities inputs. . . . . 94
3.2.12.2 TIS utilities processing. . . . . 95
3.2.12.3 TIS utilities outputs . . . . . 95
3.2.13 JBDRIVER DR11-B device driver . . . . . 95
3.2.13.1 JBDRIVER inputs . . . . . 95
3.2.13.2 JBDRIVER processing . . . . . 97
3.2.13.3 JBDRIVER outputs. . . . . 98
3.2.14 Special requirements. . . . . 99
3.2.14.1 Human performance . . . . . 99
3.2.14.1.1 TCL user interface. . . . . 105
3.2.14.1.2 Functional role of TIS users. . . . . 109

TABLE OF CONTENTS
(Continued)

3.3 Adaptation . . . . . 110
3.3.1 General environment. . . . . 110
3.3.2 System parameters. . . . . 114
3.3.3 System capacities. . . . . 115

4. QUALITY ASSURANCE PROVISIONS. . . . . 116
4.1 Introduction . . . . . 116
4.2 Test requirements. . . . . 116

5. PREPARATION FOR DELIVERY. . . . . 117
5.1 VMS command files for installation of the TIS CPCI . . . . . 117

6. NOTES . . . . . 117
6.1 AN/GYK-12 I/O interface. . . . . 117
6.1.1 AN/GYK-12 modes of input-output operation. . . . . 122
6.2 Timing tests of TIS CPCI design concepts . . . . . 125

# LIST OF FIGURES

1. Block Diagram of TIS External Interfaces. . . . . 15
2. Exchange of ICE Packets by the PSS and TIS. . . . . 27
3. Exchange of Character Data Buffers by the PSS and the TIS . . . . . 28
4. SDSS VAX-11/780 Computer System Configuration . . . . . 45
5. DR11-B Registers. . . . . 50
6. Physical Interfaces between the DR11-B, SSIM, and the RTE IOX Channel . . . . . 56
7. TIS Functional Flow Block Diagram . . . . . 59
8. Example of TIS Catalog Listing. . . . . 106
9. AN/GYK-12 Input-Output Exchanges. . . . . 118
10. AN/GYK-12 Input-Output Exchange Signals. . . . . 119
11. Control Sequences Used in AN/GYK-12 Peripheral Control . . . . . 123

# LIST OF TABLES

I. Major PSS Functions and Computer Program Modules. . . . . 17
II. Summary of PSS Program Module Functions. . . . . 18
III. PSS Functional Requests to the TIS CPCI . . . . . 22
IV. Format of ICE Packet . . . . . 29
V. Usage of Fields in ICE Packets. . . . . 31
VI. Open Command ICE Extension . . . . . 35
VII. Delete Command ICE Extension. . . . . 37
VIII. Status Return codes from the TIS CPCI to the PSS . . . . . 38
IX. DRDB Register for TIS Initiated Operations . . . . . 40
X. Theoretical Maximum Throughput Rates of VAX-11/780 Device Buses . . . 47
XI. DR11-B Command and Status Definition . . . . . 51
XII. Functions of SSIM Signal Lines to the DR11-B Interface. . . . . 57
XIII. Functional Flow Table. . . . . 60
XIV. TAPECNTRL Queue — I/O Functions. . . . . 82
XV. Initial Load Module Format . . . . . 84
XVI. Equivalence of SPS and TIS Commands . . . . . 101
XVII. Equivalence of MUFD and TIS Commands . . . . . 104
XVIII. TIS Commands Not Supported in Batch Mode. . . . . 108
XIX. TIS Commands Not Supported for PSS Users. . . . . 111
XX. Format of a Single PSSMUFD Entry . . . . . 112
XXI. Device Control Formation by Use of the Information Lines. . . . . 121

# 1. SCOPE

1.1 Identification. This ~~part of this~~ specification establishes the requirements for performance, design, test, and qualification of a computer program identified as the Tactical Interface System (TIS), SDSS-MMP-B1. The purpose of this program is to interface the Software Development Support System (SDSS) to the Microprogrammable Multiprocessor (MMP) system emulating an AN/GYK-12 executing the Programming Support System (PSS) software. The AN/GYK-12 PSS normally executes in combination with the TACFIRE Smart Peripheral System (SPS) software. The TIS Computer Program Configuration Item (CPCI) shall provide the software processing necessary to interface the PSS to the SDSS VAX-11/780, and to allow use of the PSS facilities by SDSS VAX-11/780 user personnel both interactively and in batch mode. The TIS CPCI shall execute under the control of the SDSS VAX-11/780 Virtual Memory System (VMS) operating system.

1.2 Functional summary. The TIS CPCI allows SDSS users to use the job execution facilities of the PSS software. These PSS jobs are executed in a distributed environment since the PSS executes on the MMP or AN/GYK-12, with the necessary functions being performed by the TIS software executing on the SDSS to support PSS requests. This section summarizes these functions and the user interface provided by the TIS CPCI.

The TIS CPCI shall queue user job streams to be submitted to the PSS on disk. PSS jobs are entered by VMS batch entry facilities or interactively by user-created files for execution by the PSS. On request from PSS, the TIS CPCI shall read a job from disk and transfer the associated data through the Support System Interface Module (SSIM) physical interface to PSS.

The TIS CPCI shall maintain the PSS on-disk catalog structure and access the PSS database on request from PSS. The TIS CPCI shall process the TACFIRE PSS database such that direct file interchange, between the TACFIRE PSS database and the SDSS, is accomplished without conversion by the SDSS or SPS users.

Job output sent by PSS to the TIS shall be processed for printing or punching on the SDSS VAX-11/780 peripherals. The TIS CPCI will use the spooling facilities of the SDSS VMS operating system to provide the print spooling functions. The TIS CPCI shall maintain an internal queue for punch output. Since the SDSS hardware configuration does not currently include a cardpunch peripheral, the TIS CPCI shall create the punch queue but will not punch cards directly from the queue.

The TIS CPCI shall maintain the AN/GYK-12 System File and provide the capability to downline load an AN/GYK-12 system upon command from the SDSS VAX-11/780.

The TIS CPCI shall support one real or emulated AN/GYK-12 system executing the PSS software or two real or emulated AN/GYK-12 systems executing the PSS software simultaneously.

Functional compatibility with the existing TACFIRE PDP-11 Smart Peripheral System shall be maintained by the TIS CPCI except when constrained by the environmental differences that exist between the SDSS VAX-11/780 TIS running under the VMS, the TACFIRE PDP-11/70 SPS running under the Interactive Applications System (IAS) operating system, and the TACFIRE PDP-11/35 SPS running under the RSX-11D operating system.

The TIS CPCI shall provide a command language interface for the TIS Manager and PSS users to control execution of the TIS, manipulate the job queues, control job execution, maintain the PSS master user file directory (PSSMUFD) file, show PSS job status, and show the status of the TIS CPCI. User control and query capabilities for the print queue will be implemented through the VMS command language.

The TIS CPCI shall provide a device driver to service the SSIM physical interface between the SDSS and an AN/GYK-12 Input-Output Exchange (IOE) or the IOE-extended (IOX) channel.

The TIS CPCI shall maintain error and history data logs, and shall provide error processing, history monitoring, and performance monitoring facilities to protect system integrity and to monitor the performance of the TIS CPCI.

This specification defines, in Section 3.1, the TIS interfaces in a hierarchial manner, from the system level to the physical level, and describes the hardware and software environment in which the TIS CPCI will execute and the effect of this environment on TIS. Section 3.2 provides a functional flow and a functional decomposition of the TIS CPCI. For each function, the inputs, processing, and outputs are defined. The user interface is discussed in Section 3.2. Section 3.3 provides a description of the TIS database, and the PSS database which the TIS CPCI will process. Section 4 specifies the verification methods for the functional requirements specified in Section 3. Section 5 describes the VMS command procedures (command files) that will be provided upon delivery of the TIS software. Section 6 contains a description of the interface between the SSIM electronics and the AN/GYK-12 IOE/IOX, and results of timing tests of the principal TIS software architecture concepts.

This specification is intended to provide a complete system-level design baseline from which detailed design of the TIS CPCI can proceed.

## 2. APPLICABLE DOCUMENTS

2.1 Government documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

SPECIFICATIONS:

MIL-S-83490, Military Specification for Specifications, Types and Forms, dated 30 October 1968.

AN/GYK-12 Computer Principles of Operation Manual, Programming Support System, USACSCS-TF-4-3, 29 AUG 1977.

Basic User's Guide for the Fundamental Interactive Terminal System in the Smart Peripheral System, DRAFT, TELOS Computing Inc., PM TACFIRE/FATDS, 25 MARCH 1981.

Control Data 5600 Microprogrammable 560126A Processor (RTE Processor) Manual, APR 1974.

Control Data 5600 Microprogrammable Processor 560520A (MP-60 CPU/AN/GYK-12 Emulator), CDC Publ. No. 14540600, 31 OCT 1979.

CPCEI Specification for Programming Support System Operating System Program, Litton DSD, Specification No. EL-CG-00043081, Doc. No. 595904-600, 6 JAN 1977.

CPCEI Specification for Programming Support System Programming Aids, Litton DSD, Specification No. EL-CG-00043088, Doc. No.595946-600, 12 AUG 1976.

CPCEI Specification for Smart Peripheral System, Litton DSD, Specification No. EL-CG-00043089, Doc. No. 595950-600, 6 JAN 77.

Design Description Document for Smart Peripheral System Phase 3 for TACFIRE Fire Direction System, Artillery AN/GSG-10(V), Litton DSD, Doc. No. 595950-900, 20 JUN 1976.

Development Specification for the System Interface Unit (SIU) for TACFIRE Advanced Training Program, Litton DSD, Doc. No. 114362-600, 25 JUNE 1979.

General Specification for Device-to-Computer Interface for the AN/GYK-12 Computer, Litton DSD, Doc. No. 587650-620, Rev. B, 15 FEB 1974.

LSS Assembly Language Reference Manual for the Air Defense Guided Missile AN/TSQ-73, Litton DSD, 137242-915, Rev. A, 15 SEPT 1977.

LSS Users Manual for the Air Defense System, Guided Missile AN/TSQ-73, Litton DSD, 137242-901B, Rev. B, 21 FEB 1978.

Microprogrammable Processors (5600 Series) Reference Manual, CDC Publ. No. 1423200, Rev. C., MAR 1979.

MP-60 Computer System Family Reference Manual, CDC Publ. No. 14306500, 31 MAR 1979.

MP-60 Computer System Peripheral Equipment Reference Manual, CDC Publ. No. 14063900, June 1977, Rev. B.

MP-60 Computer System TACFIRE Emulation System Reference Manual, CDC Publ. No. 11305-01A, 10 MAY 1977.

MP-60 Computer System (TACFIRE Emulation System) Reference Manual (MPX/OS), CDC Publ. No. 14696700, OCT 1980.

MPX/OS Reference Manual, CDC Publ. No. 10817300, March 1980, Rev. D.

Part II CPCEI Specification for PSS Operating System Program, EXEC, Litton DSD, Specification No. EL-CS-00043081, Vol. 3, Doc. No. 595904-650-3, July 1978, Rev. E with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, SUP3, Litton DSD, Specification No. EL-CS-00043081, Vol. 4, Doc. No. 595904-650-4, July 1978, Rev. D with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, BOOTCOPY, Litton DSD, Specification No. EL-CS-00043088, Vol. 52, Doc. No. 595905-650-52, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, INITGEN, Litton DSD, Specification No. EL-CS-00043088, Vol. 59, Doc. No. 595905-650-59, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, STPCOPY, Litton DSD, Specification No. EL-CS-00043088, Vol. 62, Doc. No. 595905-650-62, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, STPGEN, Litton DSD, Specification No. EL-CS-00043088, Vol. 63, Doc. No. 595905-650-63, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, STPBLOCK, Litton DSD, Specification No. EL-CS-00043080, Vol. 64, Doc. No. 595905-650-64, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Operating System Program, STPINFO, Litton DSD, Specification No. EL-CS-00043088, Vol. 65, Doc. No. 595905-650-65, July 1978, Rev. N/C with SCN F1.

Part II CPCEI Specification for PSS Programming Aids, Litton DSD, Specification No. EL-CS-00043088, Doc. No. 595905-650, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, Litton DSD, Specification No. EL-CS-00043089, Doc. No. 595950-650-1, 19 JUN 1978.

Part II CPCEI Specification for Smart Peripheral System, ASSUFD, Litton DSD, Specification No. EL-CS-00043089, Vol. 3, Doc. No. 595950-650-3, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, BLANDA, Litton DSD, Specification No. EL-CS-00043089, Vol. 4, Doc. No. 595950-650-4, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, FINCAT, Litton DSD, Spec. No. EL-CS-00043089, Vol. 5, Doc.No. 595950-650-5, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, HAROLD, Litton DSD, Spec. No. EL-CS-00043089, Vol. 7, Doc. No. 595950-650-7, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, ISIS, Litton DSD, Spec. No. EL-CS-00043089, Vol. 8, Doc. No. 595950-650-8, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, PRUNES, Litton DSD, Spec. No. EL-CS-00043089, Vol. 10, Doc. No. 595950-650-10, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, SID, Litton DSD, Spec. No. EL-CS-00043089, Vol. 11, Doc. No. 595950-650-11, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, SKNCAT, Litton DSD, Spec. No. EL-CS-00043089, Vol. 12, Doc.No. 595950-650-12, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, TTYNOO, Litton DSD, Spec. No. EL-CS-00043089, Vol. 14, Doc. No. 595950-650-14, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, ROLLER, Litton DSD, Spec. No. EL-CS-00043089, Vol. 15, Doc. No. 595950-650-15, 1 JUL 1978.

Part II CPCEI Specification for Smart Peripheral System, POPTOP, Litton DSD, Spec. No. EL-CS-00043089, Vol. 17, Doc. No. 595950-650-17, 1 JUL 1978.

PSS-B User/Operator Manual, Volume I, Basic Functions, Litton DSD, Specification No. 586000-906.

PSS-B User/Operator Manual, Volume II, Compiler/Assembler, Litton DSD, Specification No. 586000-904.

PSS-B User/Operator Manual, Volume III, System Generation, Litton DSD, Specification No. 586000-903.

PSS-B User/Operator Manual, Volume VII, Utilities, Litton DSD, Specification No. 586000-919.

System Specification for Programming Support System (PSS) Litton DSD, Specification No. EL-CP-00043000, 9 AUG 1976.

Test Plan for the Verification Test of the Smart Peripheral System and the TACFIRE Programming Support System (V4.2), TELOS Computing Inc., Doc. No. TCIFS-81-118, dated 3 FEB 1981.

Version Description Document, Programming System, Preliminary Version 04.2, TELOS Computing Inc., Doc. No. 591411-9042, dated 24 MAR 1981.

STANDARDS:

MIL-STD-480, Configuration Control — Engineering Changes, Deviations, and Waivers, dated 30 October 1968.

MIL-STD-483, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs, dated 31 December 1970.

MIL-STD-490, Specification Practices, dated 18 May 1972.

OTHER PUBLICATIONS:

Emulation Capabilities of a Microprogrammable Multiprocessor System, Teleprocessing Design Center, CENTACS, Ft. Monmouth, N.J., APRIL 1976.

Emulation of Tactical Data Systems in the Teleprocessing Design Center, ARTADS, Ft. Monmouth, N.J., undated.

2.2 Non-Government documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

SPECIFICATIONS:

None.

STANDARDS:

None.

OTHER PUBLICATIONS:

DR11-B/DA11-B Manual, Digital Equipment Corporation, Doc. No. EK-DR11B-TM-004, dated September 1974.

DR11-B/DA11-B Interface User's Manual, Digital Equipment Corporation, Doc. No. EK-DR11B-OP-001, dated October 1976.

VAX-11 Run Time Library Language Support Procedures Reference Manual, Digital Equipment Corporation, Doc. No. AA-J107A-TE, dated June 1981.

VAX-11 MACRO Language Reference Manual, Digital Equipment Corporation, Doc. No. AA-D032C-TE, dated March 1980.

VAX-11 Linker Reference Manual, Digital Equipment Corporation, Doc. No. AA-D019B-TE, dated March 1980.

VAX/VMS System Services Reference Manual, Digital Equipment Corporation, Doc. No. AA-D018B-TE, dated March 1980.

VAX/VMS I/O User's Guide, Digital Equipment Corporation, Doc. No. AA-D028-TE, dated September 1980.

VAX-11 Guide to Creating Modular Library Procedures, Digital Equipment Corporation, Doc. No. AA-H500B-TE, dated April 1980.

VAX-11 Record Management Services User's Guide, Digital Equipment Corporation, Doc. No. AA-D781C-TE, dated March 1980.

VAX-11 Record Management Services Reference Manual, Digital Equipment Corporation, Doc. No. AA-D031C-TE, dated March 1980.

RMS-11 User's Guide, Digital Equipment Corporation, Doc. No. AA-D538A-TC, dated March 1979.

RMS-11 MACRO Reference Manual, Digital Equipment Corporation, Doc. No. AA-H683A-TC, dated March 1979.

VAX-11/RSX-11M Programmer's Reference Manual, Digital Equipment Corporation, Doc. No. AA-D020B-TE, dated March 1980.

VAX/VMS Real-Time User's Guide, Digital Equipment Corporation, Doc. No. AA-H784A-TE, dated March 1980.

VAX/VMS Guide to Writing a Device Driver, Digital Equipment Corporation, Doc. No. AA-H499B-TE, dated March 1980.

VAX/VMS System Manager's Guide, Digital Equipment Corporation, Doc. No. AA-D027B-TE, dated March 1980.

VAX/VMS Operator's Guide, Digital Equipment Corporation, Doc. No. AA-D025B-TE, dated March 1980.

## 3. REQUIREMENTS

This section specifies the functional requirements that the TIS CPCI must satisfy. These requirements provide the baseline for formal qualification through acceptance testing.

Section 3.1 defines the TIS CPCI as a group of functions executing under the control of the VMS, the functional requirements imposed on the TIS CPCI by the PSS, the I/O requirements of the VAX-11/780 DR11-B interface, the VMS operating system interface requirements, and the VAX/VMS Record Management Services (VAX-11 RMS) interface requirements.

Section 3.2 specifies the functional flow within the TIS CPCI and details the inputs, processing, and outputs for each major function, and the TIS user interface including the TIS Command Language (TCL) syntax and grammar.

Section 3.3 contains the TIS database characteristics and defines the structure of the PSS database which the TIS CPCI shall process.

The TIS CPCI shall provide a functional and physical interface to the PSS that is compatible with the PSS/SPS interface implemented on the current TACFIRE PDP-11 support software system, that is, the TIS CPCI shall not impose any changes or new requirements upon the PSS or the AN/GYK-12 I/O interface. To accomplish this without modification of the PSS, the TIS CPCI will perform the following functional requirements, which are based on the processing currently provided by the TACFIRE PDP-11 SPS.

The TIS CPCI shall provide input queuing of PSS jobs. The PSS job class shall be checked to determine that the interfacing PSS supports the particular PSS job class. The PSS job control language (JCL) shall be validated before entry of the batch job into the PSS input queue. Facilities of the PSS JCL dependent upon processing by the SPS (for example, PSS CALL files) shall be provided by the TIS CPCI. Upon request from PSS the job shall be read from the input queue maintained by the TIS CPCI and transferred through the SSIM to the PSS. TIS shall maintain the status of the PSS job queue and shall report this status to an TIS user upon request. TIS shall provide operator control of the PSS job queue internal to the TIS, and shall provide operator control of jobs currently executing in each PSS.

The TIS CPCI shall accept job output from PSS and shall interface to the VMS print symbiont to enter the printed batch output from the PSS into the VMS

print queue. The TIS users will determine print queue status and change forms by using the standard VMS print queue maintenance commands.

For punch output from PSS jobs, the TIS CPCI shall maintain an internal queue for the PSS punched data. The TIS CPCI shall provide the user with the capability to determine the status and to control the punch output queue. In the SDSS VAX-11/780 TIS CPCI, the use of the PSS punch output option should not normally be required; however, the TIS CPCI should provide the capability to save punch output, if any, on the appropriate SDSS disk for subsequent transfer to tape.

TIS shall provide PSS file handling functions on the SDSS disks and tapes. TIS shall execute PSS requests to open, read, write, close, and delete PSS disk files. TIS shall also execute PSS requests to space forward, space backwards, and rewind a PSS disk file.

TIS shall write and read PSS tape files using the SDSS magnetic tape drive peripherals. TIS shall execute PSS requests to space forward and space backwards tape records and files, and shall rewind and unload tapes upon command from the PSS. TIS shall write tape end-of-file marks and tape end-of-volume marks, logically mount and dismount tapes, and sense write protection upon command from the PSS.

TIS shall provide a command language to the TIS Manager and the PSS users. The PSS users will have a subset of the TIS Manager command language capabilities. The command language shall provide commands to control TIS execution, submit PSS jobs, scan and validate the PSS JCL, control and query the PSS job queue and punch queue, downline load each AN/GYK-12 or MMP attached to the SDSS VAX-11/780 through an SSIM, initiate SSIM test messages, send TIS Manager messages to the PSS, and list and maintain the PSSMUFD catalog.

TIS shall downline load an AN/GYK-12 or the MMP Emulator. Upon request, TIS shall transfer the initial load module, which for the PSS contains the secondary bootstrap loader, and then shall read the AN/GYK-12 System File (upon command from the PSS secondary loader) from the SDSS disk and transfer the PSS to the MMP emulator or AN/GYK-12 computer. The TIS should have the capability to downline load a properly formatted load module created by the PSS INITGEN CPCEI, or the L-3050 Support Software (LSS) System CPCEI, in addition to downline loading a PSS load module, since these capabilities were supported by previous versions of the PDP-11 SPS.

TIS shall provide an automatic or TIS Manager-initiated test of the SSIM. The TIS Manager shall have the capability to specify if the SSIM test is to execute automatically or execute upon manual initiation. If automatic execution is selected, the TIS Manager will be able to specify the time interval between SSIM test messages.

The TIS CPCI shall control the operation of one or two Digital Equipment Corporation (DEC) DR11-B direct memory access (DMA) interfaces to perform I/O between the SDSS VAX-11/780 and each AN/GYK-12 Input Output Unit (IOU) or the MMP IOU emulation. The DR11-B device driver shall initially support one or two AN/GYK-12 or MMP computers, each communicating through an SSIM and executing a PSS, simultaneously. The DR11-B device driver shall be written in accordance with the VAX/VMS Guide to Writing a Device Driver (AA-H499B-TE). Reconfiguration of the DR11-B device driver to support additional DR11-B to SSIM interfaces should be accomplished through assembly of the driver source.

The TIS CPCI shall control resource allocation to prevent or detect deadlocks between PSS jobs executing on one or two AN/GYK-12 or MMP computers. The allocation of tapes and disk files shall be controlled between PSS jobs, possibly executing under different PSS computers, so that a PSS job will not be aborted due to allocation of a tape or disk file to another PSS job.

The TIS CPCI shall have the capability to process disk packs containing a TACFIRE PSS on-disk database from the TACFIRE PDP-11 SPS without prior data conversion. TACFIRE PSS database disk packs, that are hardware-compatible with the SDSS RP06 disk drives, will be mounted directly in a SDSS RP06 disk drive specified by the TIS Manager; alternatively, the TACFIRE PSS database may be stored on any other SDSS disk drive. File transfer from the TACFIRE PSS to the SDSS TIS may require that the TIS Manager use the TIS CREATE or UPDATE command (see 3.2.14.1) to make the appropriate new entries into the PPSMUFD master catalog file for the PSS filenames.

The TIS CPCI shall have the capability to process a PSS on-disk database such that the PSS data files, subsequent to processing by the TIS, are fully compatible and interchangeable with the TACFIRE PSS and the TACFIRE PDP-11 SPS. File transfer from the SDSS TIS to the TACFIRE PSS/SPS may require that a PDP-11 SPS operator use the TACFIRE SPS MUFD CPCEI to make or change the appropriate new entries into the TACFIRE master catalog file for the PSS filenames. If the entries for the appropriate PSS filenames already exist in the TACFIRE master catalog file, then the PSS files produced by the TIS CPCI shall be fully compatible with the TACFIRE PDP-11 SPS and PSS.

PSS tapes created by the TIS CPCI shall be compatible with the TACFIRE PDP-11 SPS. The TIS CPCI shall process PSS tapes created by the TACFIRE PDP-11 SPS.

The TIS CPCI shall provide on-disk database purge mechanisms necessary for declassifying SDSS VAX-11/780 disk packs containing national security classified data up to and including the SECRET (collateral) level.

The TIS CPCI shall provide error and transaction history data logging to monitor the performance and integrity of the SDSS TIS software and the PSS database. The TIS Manager shall have the capability to control the error and transaction history data logging, and to display the logging results.

3.1 Computer program definition. The TIS CPCI will execute as a group of applications programs or detached processes (except the DR11-B driver) on the SDSS VAX-11/780 running under the control of VMS. A process is the basic entity created and scheduled by VMS that provides the context in which an image executes. A process consists of address space, hardware context, and software context. The processes of the TIS group will execute within the SDSS VAX-11/780 VMS with a group user identification code (UIC) of [300,300].

The TIS DR11-B device driver will be installed as part of VMS. The TIS DR11-B device driver will be referenced by the VMS as JBDRIVER, and the TIS software will access the JBDRIVER as devices JBA0 and JBA1. Each separate DR11-B device will be treated as a separate controller on the same SDSS VAX-11/780 Unibus Adapter (UBA).

The SDSS VMS System Manager will have control of the execution priority, quotas and other VMS resources required by the TIS CPCI.

The TIS software will execute in a group-privileged mode under VMS. The detached processes of the TIS CPCI shall affect or control only member processes of the TIS group. The TIS CPCI group of detached processes shall be unique within the SDSS VAX-11/780 VMS, so that the TIS processes cannot affect other detached processes foreign to the TIS CPCI.

TIS may require other VMS privileges in addition to normal privileges, such as OPER, to perform required functions. TIS shall not require privilege to interfere with normal VMS operation (ALTPRI), to compromise file security (SYSPRV or BYPASS), or to directly address executive space (CMKRNL).

The TIS CPCI may require amounts of resource quotas (for example, asynchronous system trap (AST) Queue Limit or Open File Limit) in addition to the usual SDSS VMS defaults defined in the SDSS User Authorization File (UAF).

The TIS CPCI shall not require permanently dedicated peripherals (for example, permanently dedicated disk drives, printer, or card reader) in the SDSS VMS environment.

The TIS CPCI will be queue-driven. There are two general types of queues that the TIS CPCI will maintain:

a. PSS Interface Control Element (ICE) packets.

b. PSS job-oriented queues.

As multiple PSS ICE packets are sent by the PSS to be processed by the TIS CPCI, the ICE packets for each function required by the PSS (see 3.1.1.2.1) will be maintained in FIFO-ordered queues. PSS ICE requests that involve a Queue-I/O or VAX-11 RMS I/O operation to a tape or disk file will be initiated

by the TIS software and subsequently, upon AST or event flag completion notification, be checked for completion status. Thus, the design of the TIS software will not be single-threaded but will allow multiple I/O requests to be outstanding and in progress to service multiple PSS ICE requests, and the order of completion of the PSS ICE requests will be determined by the order of completion on the SDSS. The TIS processes will hibernate, to reduce computer resource contention, if the queue being serviced is empty. AST notification of a pending ICE packet or TIS interprocess request packet will cause a wakeup to be scheduled for the appropriate TIS processes. TIS processes to be wakened externally will be identified by the VMS process identification number.

The queue lengths for PSS requests (for example, the number of PSS ICE packets for which processing has not yet been initiated by the TIS software) will be variables that can be observed by the TIS Manager through the TCL SHOW command (see 3.2.14.1).

The PSS job-oriented inputs will be serviced as multiple job slots by the TIS CPCI. The TIS software will dequeue PSS jobs from the TIS Input Queue based upon the frequency of requests to read the Input Queue from the PSS. The TIS software will allow the PSS to process multiple jobs with one job allocated to each active PSS job slot, and the TIS software will monitor the status of the multiple PSS jobs. The TIS software will support requests from the PSS for multiple batch jobs, with the number of jobs in progress a direct function of the capabilities of the PSS and the allowable amounts of resources available from the SDSS VAX-11/780 VMS.

The TIS Manager shall have the capability, during the TIS startup and initialization sequence, to interactively configure the TIS CPCI to support either one or two active AN/GYK-12 Programming Support Systems. The TIS Manager shall have the capability to downline load one AN/GYK-12 or MMP while the other AN/GYK-12 or MMP is executing a PSS.

The TIS CPCI shall initially contain sufficient queue and data storage to process up to 72 separate PSS files. Each PSS File Descriptor Table (FDT) can contain up to 144 entries; for comparison with the TIS CPCI, the TACFIRE PDP-11 SPS software file buffer capacity currently allows up to 50 separate PSS files (25 for each possible PSS).

If the initial number of open PSS files is exceeded, the TIS CPCI shall extend the queue and data storage to support up to an increment of n additional separate PSS files, where n is a variable internal to the TIS, but controllable and alterable by the TIS Manager through the TIS SET command. The value of n will be in the range $0 \leq n \leq 72$ for a dual or single PSS configuration.

If the extension, n, beyond the open PSS file threshold is zero or is not sufficient to support all of the PSS file requests, the TIS CPCI shall

throttle and control subsequent PSS file open requests from other PSS jobs such that currently active PSS jobs may complete execution.

The TIS CPCI, consisting of a group of detached, group-privileged VMS processes, will be designed to use four primary VMS capabilities to allow the processes in the TIS group to share data, control mechanisms, and to service queues of ICE packets from the PSS:

a. A group-global common area, TISGBL.

b. A set of group-global event flags.

c. Interprocess communication through the VMS Mailbox facility.

d. A group-shared run-time library, TISRTL.

The TIS global data area, TISGBL, will contain the data in the TIS software that must be globally accessible by different processes in the TIS group. The TISGBL will include batch input and output queue structures, variables that affect TIS processing and that are interactively alterable and displayable by the TIS Manager and PSS users, the ICE queues that are input to each TIS process that services PSS requests, and other control structures such as the TIS file access tables. The TISGBL will be a temporary global section and will be deleted during the TIS termination sequence. The TISGBL will be periodically paged to disk through the VMS Update Section system service as a part of the System Initiator/Terminator (SYSINITRM) function. The initial size of the TISGBL will vary according to whether a single or dual PSS configuration is to be supported.

Group-global event flags will be used to signal conditions, such as termination, request in progress, or other status indicators, among the TIS processes. The use of group-global event flag clusters will be limited to indicate the reason for process activation; other necessary input data to each TIS process will be accessed through the TISGBL and the PSS database. The TIS group-global event flag clusters will be temporary and will be deleted during the TIS termination sequence.

In addition to TIS interprocess communication through event flags and the TISGBL, TIS interprocess communication will also use the VMS Mailbox system service. A Mailbox is a record-oriented virtual I/O device created by a process. Other processes may write data to a Mailbox by queueing write requests, or may read messages in a Mailbox by queueing read requests. A process can request AST notification when data is written to one or more Mailboxes by an external process. TIS Mailboxes will be temporary and will be deleted during the TIS termination sequence.

The TIS run-time library (TISRTL) will contain a set of frequently used common procedures. The TISRTL will include PSS data management, buffer ser-

vice, history and error logging, monitoring, and ICE status and error checking procedures. The TISRTL shall be written in accordance with the VAX/VMS Guide to Creating Modular Library Procedures (AA-H500B-TE).

The TIS processes will interface with VMS (including the DR11-B driver) through the Queue-I/O system service and calls to the VMS Symbiont Manager, and through other VMS system services as necessary to implement TIS group functions. The TIS database and the PSS database will be accessed through VAX-11 RMS.

3.1.1 Interface requirements. This section discusses the interface requirements imposed on the TIS CPCI by other computer programs and equipment. These requirements include internal and external interfaces to the SDSS. The external interface requirements defined in this section are imposed by the PSS and the I/O characteristics of the DR11-B. The internal requirements defined in this section are imposed by the VAX-11/780 VMS, the VAX/VMS Record Management Services (VAX-11 RMS), and the DR11-B device driver.

The TIS CPCI shall not impose any new requirements or changes upon the PSS interface, that is, the TIS CPCI shall use the same link-level interface mechanism and logical I/O procedures as the PDP-11 SPS/PSS.

The TIS CPCI shall operate with one or two DR11-B devices installed in the SDSS VAX-11/780 and with the DR11-B device driver incorporated into the VMS.

3.1.1.1 Interface block diagram. Figure 1 shows an overview of the TIS external interfaces.

The TIS CPCI interfaces at the link-level and the logical I/O level with the PSS to perform functions in support of the PSS. These functions are initiated by fixed length, preformatted Interface Control Element (ICE) packets. To transfer the ICE and data required by the functional interface in a controlled manner, the TIS CPCI and the PSS must follow the SPS/PSS interface protocol as defined in 3.1.1.2.1 and other supporting documents.

The physical I/O interface between the AN/GYK-12 Input Output Exchange (IOE) channel and the VAX-11/780 DMA DR11-B interface is provided by the SSIM sequence and control logic implemented in the SSIM electronics.

The TIS CPCI interfaces to the DR-11B Device Driver (JBDRIVER) through the VMS Queue I/O System Service. TIS also interfaces with VMS for process scheduling, intertask communications, print spooling, and resource control. The TIS CPCI interfaces with the VAX/VMS Record Management Services to access disk files. These external interfaces are detailed in the following sections.

3.1.1.2 Detailed interface definition. This section specifies, in detail sufficient to permit design of the TIS CPCI, the functional relationship of

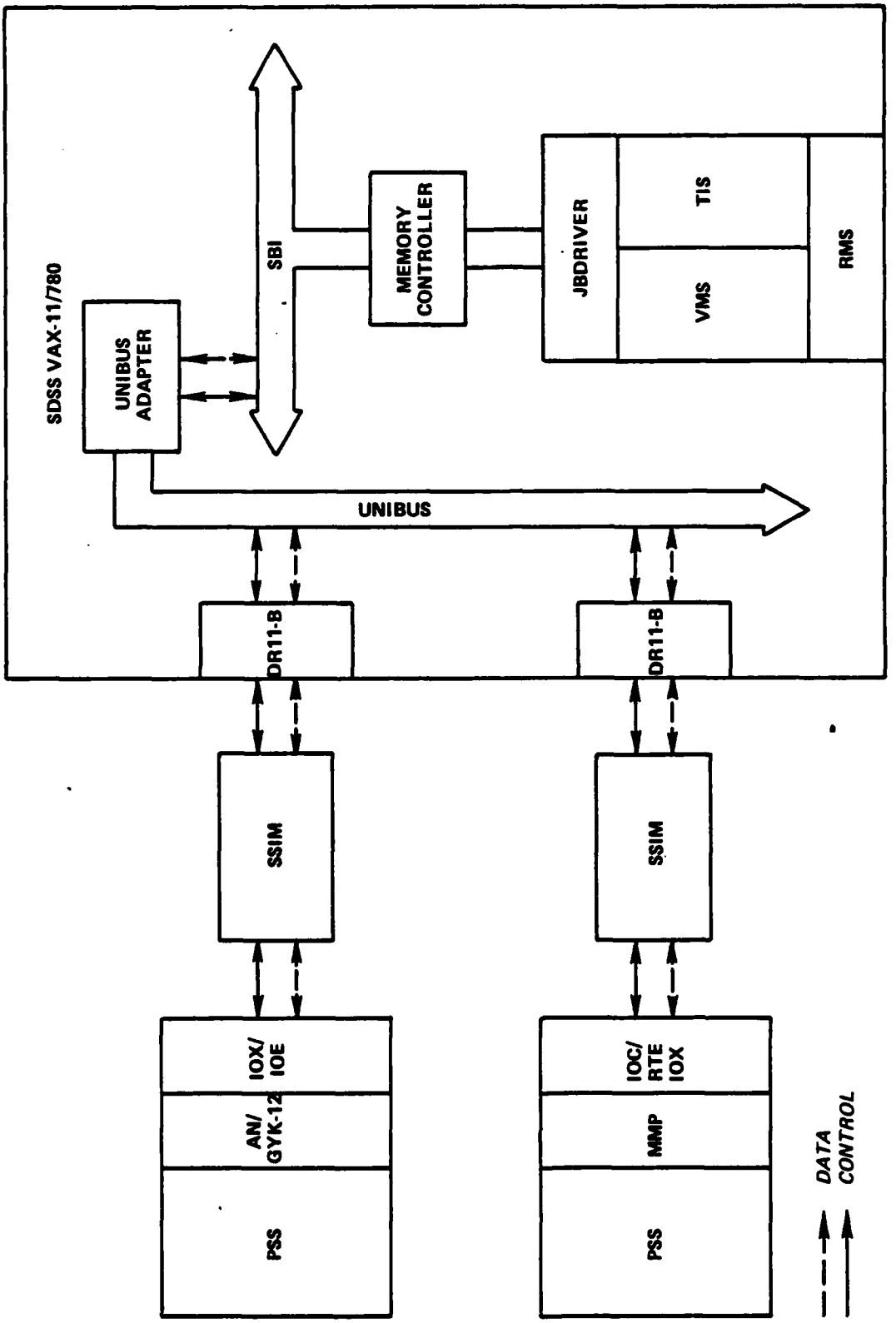



FIGURE 1. BLOCK DIAGRAM OF TIS EXTERNAL INTERFACES

the TIS CPCI to the PSS and the SDSS VMS, and the interface between the TIS CPCI and the PSS, consisting of the PSS ICE packets received and transmitted through the SSIM, that must be processed by the TIS CPCI. The interfaces between the VMS operating system and the TIS CPCI, and the interfaces between the VAX/VMS Record Management Services and the TIS CPCI, are also specified.

3.1.1.2.1 PSS interface. The PSS, executing on the AN/GYK-12 or MMP and communicating with the TIS CPCI through an SSIM, controls the sequence of PSS program module execution, allocates and deallocates PSS internal resources, and services all AN/GYK-12 or MMP peripheral devices for the programs executing under the control of the PSS. Requests for I/O services, calls from one PSS program module to another (global calls), requests for timer queue entries, memory storage paging, and all other aspects of the software context for a PSS applications program are provided by the PSS operating system.

The PSS consists of 11 major functions and 24 major program modules as summarized in table I. Each major program module performs a number of PSS functions; these functions are summarized in table II.

The PSS is a general purpose computing system that aids in the development of tactical data systems. The PSS supports compilations, application program debugging and testing, configuration management, system generation, and utility functions. The PSS shall operate in combination with the TIS CPCI, which shall provide access to disk data files and commercial peripherals for the PSS.

Using the SDSS and the TIS CPCI, the PSS Tactical Procedure Oriented Language (TACPOL-B) compiler compiles source programs from the SDSS VAX-11/780 disks and stores the object programs on the SDSS VAX-11/780 disks, and the PSS COMPOOL generator retrieves COMPOOL source from the SDSS VAX-11/780 disks and stores COMPOOL object on the SDSS VAX-11/780 disks. PSS system generation programs use the object modules as input to the system generation process, and the PSS loader retrieves object programs from the SDSS VAX-11/780 disks for testing.

The TIS CPCI shall provide simultaneous access to the input streams and output streams of multiple PSS jobs by queueing job input streams first-in first-out (FIFO) for execution on the host AN/GYK-12; by collecting the job output from the PSS; and at PSS job termination, printing the contents of the job output stream.

The TIS CPCI shall substitute for the TACFIRE PDP-11 SPS and shall perform, as a minimum, the functions that the PSS must ordinarily issue to the TACFIRE PDP-11 SPS.

The TIS CPCI shall support asynchronous I/O between the PSS and the SDSS VAX-11/780, utilizing the SSIM to provide fast response to PSS user requests.

Table I. Major PSS Functions and Computer Program Modules

| FUNCTION | NAMES OF MAJOR PSS PROGRAM MODULES |
|---|---|
| EXECUTIVE | EXECUTIVE, SUP3, SR4 |
| FILE MANAGEMENT | SUP2, SETALIAS, DCLFILE, DMSUP2B, RSUP2, GETFILE |
| SALVAGE POINT RECORDING | CHKPOINT |
| INITIALIZATION | SUP9, SUP9LDR, SUP9A, FACA |
| OPERATOR COMMUNICATIONS | SUP11, OPERIN, ATCONV, OPEROUT |
| INPUT STREAM INTERPRETER | SUP10 |
| SYSTEM DUMP | OSDUMP |
| SYSTEM PATCH | TPATCH |
| PROGRAM LOADER | TPBUILD |
| INPUT STREAM READER | CARDIN |
| OUTPUT STREAM WRITER | PRINT |

Table II. Summary of PSS Program Module Functions

| PROGRAM MODULE | MAJOR FUNCTIONS PERFORMED IN THE PSS |
|---|---|
| EXECUTVE | Exercises overall control of the AN/GYK-12 CPU, performs resource allocation, schedules job execution, and processes error interrupts. The EXECUTVE receives and processes all service requests, and internally handles all non-I/O requests. |
| SUP3 | Receives and interprets requests for I/O services and I/O interrupts. SUP3 controls all peripheral device interfaces. |
| SR4 | Performs interrupt processing and contains the PSS idle loop. |
| SUP2 | Processes all file manipulation requests, except core and random access auxiliary memory (RAAM) files, and file close requests. |
| SETALIAS | Creates an alias for a named file description and links the alias to an existing file description. |
| DCLFILE | Creates a new file description and modifies an existing file description. |
| DMSUP2B | Performs open file processing for all files including core and RAAM files. |
| RSUP2 | Processes all requests for addition, deletion, modification, or access to RAAM files. |
| GETFILE | Retrieves a copy of a file description. |
| CHKPOINT | Records or restores a copy of the TACFIRE Division or Battalion database for use by an applications program executing in the PSS. |
| SUP9 | Initializes the PSS memory map, job control tables, I/O access tables, and memory management tables. |

Table II. Summary of PSS Program Module Functions (Continued)

| PROGRAM MODULE | MAJOR FUNCTIONS PERFORMED IN THE PSS |
|---|---|
| FACA | Reads the PSS load file, unpacks the physical records, and loads the records into the PSS memory. |
| SUP11 | Receives all PSS operator input messages and processes messages addressed to the system. SUP11 generates status, error condition, and hardware malfunction reports. |
| OPERIN | Routes operator input messages addressed to user jobs. |
| ATCONV | Performs character conversions between ASCII and typewriter communication code (IBM SELECTRIC). |
| OPEROUT | Generates the I/O requests necessary to output a message to the PSS operator. |
| SUP10 | Processes JCL card images to create and terminate jobs, to execute programs, and to establish or modify file descriptions. SUP10 supervises the execution of a job in the PSS, collects and displays job statistics, and processes abnormal job termination. |
| OSDUMP | Prints selected portions of AN/GYK-12 memory. |
| TPATCH | Processes JCL requests to modify or dump programs, Communications Pool (COMPOOL) pages, core pages, RAAM bands, and direct access file records. |
| TPBUILD | Converts an object module to a load module that is executable and includes the new program in the PSS list of executable programs. |
| CARDIN | Reads card-image records from the logical input spool file for each job. |
| PRINT | Writes printline-image records and card-image records to the logical output spools for each job. |

The PSS may make multiple requests of the TIS CPCI; the order of response to the PSS requests by the TIS CPCI shall depend upon the order in which the requests are completed in the SDSS VAX-11/780. While one PSS request is being serviced by the TIS CPCI, several more requests may be sent to the TIS from the PSS. The order of completion for the PSS requests is not necessarily the same as the order in which the requests were sent by the PSS to the TIS CPCI.

The PSS communicates with the TIS CPCI, through the SSIM, by using a fixed-length, 16-byte, preformatted ICE packet. The TIS CPCI shall perform all processing for the PSS based upon the contents of the ICE packet control group.

To support the asynchronous I/O, at least two physical I/O operations may be necessary to satisfy a request from the PSS to the TIS CPCI.

The first physical I/O operation sends an ICE packet control group from the PSS to the TIS CPCI. The ICE packet, which signals the beginning of a logical PSS request, indicates the following information to the TIS CPCI:

a. Request type.

b. Length of the actual data buffer that is to be transferred between the PSS and TIS CPCI computers.

c. The job slot number of the PSS job making the request to the TIS CPCI.

d. A unique file number, referring to a specific file entry in the PSS FDT, that identifies a specific file in the PSS on-disk database maintained by the TIS CPCI on the SDSS VAX-11/780 disks. The file number refers to the same file in all file-oriented ICE requests for a given file.

e. A queue table number that uniquely identifies to the host PSS the logical request and the TIS CPCI response to the PSS logical request.

After the TIS CPCI processes the ICE packet, the TIS CPCI shall then signal the PSS to begin the second physical I/O operation that is the subject of the information contained in the ICE packet, by placing the PSS I/O queue table number in the SSIM DR11-B data buffer register (DRDB) and causing the PSS to be interrupted on channel $17_8$ (attention interrupt).

The second physical I/O operation may transmit the actual data, indicated by the information in the corresponding ICE packet, through the SSIM. For read or write operations, the data consists of the appropriate data record; for open file or delete file requests, the second I/O operation consists of a fixed-length string which further identifies the file to be processed by the TIS CPCI.

In the case of PSS logical operations that require a response indicating to the PSS that the TIS CPCI has completed the requested operation, a Status ICE packet control group shall be sent from the SDSS VAX-11/780 to the PSS. The Status ICE packet contains information that identifies the request to the PSS and the TIS completion status of the TIS CPCI.

If an error occurs on the TIS CPCI processing of a PSS request, an Error ICE packet control group shall be sent from the SDSS VAX-11/780 to the PSS. The Error ICE packet contains information that identifies the request to the PSS and the TIS error status of the PSS request.

3.1.1.2.1.1 Functional description of PSS interface. This section provides a description of the functions that the PSS may request of the TIS CPCI and the functions that the TIS CPCI may request of the PSS.

The PSS functions that the PSS may request of the TIS CPCI are listed in table III, and whether or not a Status ICE packet must be returned by the TIS CPCI is indicated for each functional request. An Error ICE packet may be returned to the PSS by the TIS CPCI whenever an error is detected during TIS processing of any of the PSS functional requests.

For the Open PSS disk file function, the PSS can open a new or existing disk file on the SDSS VAX-11/780 disks. The PSS specifies the filename, member name, version number, device type (only disk is supported by the ICE), the initial file access conditions (exclusive use of an existing file, shared use of an existing file, or create a new file), the final file disposition (delete, keep, pass to next job, or temporary retention), the file access type (read only, read/write, update, and modify with the record pointer for the file positioned at the end of the file), the blocksize (in AN/GYK-12 full words and less than or equal to 2048), the initial file space to be allocated for a new file (in units of 512 bytes — disk sector size), the extent size for file extension (in units of 512 bytes — disk sector size), the record access type, the classification of the data (UNCLASSIFIED, CONFIDENTIAL, or SECRET), and the disk volume name for creation of a new file.

In addition, the PSS assigns a unique file number to refer to and identify each separate file in all file-oriented ICE requests. This file number corresponds to the number of the entry for the file in the PSS file descriptor table (FDT).

For the Close PSS disk file function, the PSS can close or deaccess a previously opened PSS disk file. The PSS file to be closed is uniquely identified by the file number in the ICE packet.

For the Read Open PSS disk file function, the PSS can read a record, either in sequential access mode or specified by the direct access record

Table III. PSS Functional Requests to the TIS CPCI

| FUNCTION | COMPLETION STATUS ICE REQUIRED FROM THE TIS |
|---|---|
| Open a PSS disk file | Yes |
| Close a PSS disk file | Yes |
| Read an open PSS disk file | No (Error ICE Possible) |
| Write an open PSS disk file | No (Error ICE Possible) |
| Delete a PSS disk file | Yes |
| Start job | Yes |
| Terminate job | Yes |
| Read from job input queue | No (Error ICE Possible) |
| Write to job output queue | No (Error ICE Possible) |
| Write PSS operator message to TIS | No (Error ICE Possible) |

| FUNCTION | COMPLETION STATUS ICE REQUIRED FROM THE TIS |
|---|---|
| Unsolicited operator message from TIS | No |
| Error Event | No |
| Test SSIM message | No |
| System startup (class list) | No |
| Write tape record | No (Error ICE Possible) |
| Read tape record | No (Error ICE Possible) |
| Forward space n tape records | Yes |
| Backspace n tape records | Yes |
| Forward space n tape files | Yes |
| Backspace n tape files | Yes |

Table III. PSS Functional Requests to the TIS CPCI (Continued)

| FUNCTION | COMPLETION STATUS ICE REQUIRED FROM THE TIS |
|---|---|
| Rewind tape | No (Error ICE Possible) |
| Rewind and unload tape | No (Error ICE Possible) |
| Write EOF tape mark | Yes |
| Mount previously allocated tape | Yes |
| Release tape and dismount | Yes |
| Mount previously unallocated tape | Yes |
| Sense write protect and tape status | Yes |
| Start punch output queue | No (Error ICE Possible) |
| Write to punch output queue | No (Error ICE Possible) |

| FUNCTION | COMPLETION STATUS ICE REQUIRED FROM THE TIS |
|---|---|
| Read System File | No (Error ICE Possible) |
| Write to open PSS disk file subcommand | No (Error ICE Possible) |
| Read from open PSS disk file subcommand | No (Error ICE Possible) |
| Forward space n PSS disk file records | Yes |
| Backspace n PSS disk file records | Yes |
| Rewind PSS disk file | Yes |

number in the ICE, of a given fixed length, where the length is specified in the ICE packet. The PSS file to be read must have been previously opened, and is uniquely identified by the file number in the ICE packet.

For the Write Open PSS disk file function, the PSS can write a record, either in sequential access mode or specified by the direct access record number in the ICE, of a given fixed length, where the length is specified in the ICE packet. The PSS file to be written must have been previously opened, and is uniquely identified by the file number in the ICE packet.

For the Delete PSS disk file function, the PSS can delete a closed PSS disk file. The file to be deleted is identified by a PSS filename, member name, and version number.

The PSS requests job input and job output functions by four job control ICE packets:

a. Start PSS job. The next available entry shall be taken from the TIS input queue and written to the requesting PSS. The PSS may also request that a job output print file be opened with this function. If no job is available for PSS execution, then the TIS CPCI shall notify the PSS with a Status ICE.

b. Terminate PSS job. All files associated with a specific PSS job shall be closed by the TIS CPCI, and the output print file, if any, shall be queued for printing by the SDSS VMS. All outstanding PSS I/O requests for the specified PSS job shall be cancelled by the TIS CPCI.

c. Read from job input queue. The TIS CPCI shall write a block of 12 card images to the PSS from the JCL file for the appropriate job. If no job is available for PSS execution, then the TIS CPCI shall notify the PSS with a Status ICE. Data that is Job Control Language will be converted by the TIS CPCI from ASCII to EBCDIC prior to transmission through the SSIM to the PSS, with tabs converted to the appropriate number of EBCDIC blanks. Data that is source input to the TACPOL-B compiler will be searched for tabs, and the tabs will be converted to the appropriate number of ASCII blanks.

d. Write to job output queue. The TIS CPCI shall accept data from the PSS to be written into the output print file for a specified job. This data will be received by the TIS CPCI from the PSS in ASCII format, and the first byte of each record is assumed to be a carriage control character.

The PSS assigns a unique job slot number to refer to and identify each separate job in all job control-oriented ICE requests. This job slot number corresponds to the number of the PSS job which is currently being executed. Multiple jobs can be executed by a single PSS; the job slot number can be in the range of 1 to 15 decimal. Each ICE packet requesting the completion of a job control function for a specific job contains the job slot number.

For the Write PSS Operator Message to the TIS function, the PSS requests that the TIS CPCI accept a variable length ASCII-formatted message, with the length specified in the ICE packet, to be written to the TIS Manager terminal or, if the TIS Manager is not logged into the SDSS VMS, to be written to the SDSS master operator console (OPA0:).

The TIS CPCI shall accept a PSS operator message to be sent to the PSS from the TIS Manager. This message shall be a maximum of 80 alphanumeric and special characters.

The TIS CPCI shall initiate error messages defining error events through the Error ICE function.

The TIS CPCI shall send a Test SSIM Message ICE to the PSS and shall accept the PSS response. If the PSS response is illegal or if a response is not received within a time interval specified by the TIS Manager, the TIS CPCI shall inform the TIS Manager or the SDSS VAX-11/780 VMS operator and, if enabled by the TIS Manager, will suspend all SSIM I/O operations for the responding PSS.

The TIS CPCI shall accept a System Startup ICE from the PSS and shall subsequently read the Class List from the requesting PSS. More than one Class List may be received from a single PSS.

The TIS CPCI shall process PSS requests to manipulate physical tape. Within each ICE request type, the PSS specifies the tape unit number. For read tape and write tape, the PSS specifies the record size, and for forward space and backspace tape files and tape records the PSS specifies the number of files or records to space. The PSS may request tape rewinds, rewinds and unloads, tape mark writes, and determination of write-protect status from the TIS CPCI. The PSS must request the TIS CPCI to mount a given tape before attempting to issue tape commands to the tape. A tape mount request may be for an allocated or previously unallocated SDSS tape drive.

The TIS CPCI shall process PSS ICE requests to open a punch output file and to write to an open punch output file. A PSS punch output file shall be assigned to each requesting job slot.

The TIS CPCI shall process PSS ICE requests to manipulate disk files. Within each ICE request type, the PSS specifies the file number. The TIS CPCI shall process write, read, space forward or backspace records, and rewind functions for PSS disk files. The PSS specifies the number of records to space or the length of the record to sequentially read or write.

3.1.1.2.1.2 Description of PSS ICE packet interface. This section defines the ICE packet formats and the data to be exchanged between the TIS CPCI and the PSS for each function that the PSS may request of the TIS CPCI as specified in 3.1.1.2.1.1 and table III.

For transfers from the VAX-11/780 to the AN/GYK-12 or the MMP, the SSIM will output the low-order byte of a given data word first; this byte is then put into the high-order byte of the AN/GYK-12 destination address by the IOU. For transfers from the AN/GYK-12 or the MMP to the VAX-11/780, the IOU will output the high-order byte of a given data word first; this byte is then put into the low-order byte of the VAX-11/780 destination address by the SSIM. This procedure must be followed to maintain compatibility with the TACFIRE SPS; thus, byte-swapping of certain word items is necessary.

The PSS and the TIS CPCI shall exchange ICE packets as shown in figure 2. Due to addressing and differences in the relationship of numerical significance to bit position between the AN/GYK-12 and the SDSS VAX-11/780, the ICE packets will be stored in different physical formats in the memory of each computer. These differences in significance will cause the TIS CPCI to byte-swap AN/GYK-12 half-words to obtain the true numeric value. For character (byte) data no byte-swapping shall be necessary. Character data buffers will be stored in the memory of each computer as shown in figure 3.

The format of an ICE packet shall be as shown in table IV. The usage of each field in the ICE packet for each request type shall be as shown in table V.

To open a disk file, the PSS passes two data packets to the TIS CPCI. The first packet is the Open File ICE, with the N field giving the size (in AN/GYK-12 full words) of the second data packet. The second data packet, the Open Command ICE Extension, shall be formatted as shown in table VI and shall be read from the PSS by the TIS CPCI.

To delete a disk file, the PSS passes two data packets to the TIS CPCI. The first packet is the delete file ICE, with the N field giving the size (in AN/GYK-12 full words) of the second data packet. The second data packet, the Delete Command ICE Extension, shall be formatted as shown in table VII and shall be read from the PSS by the TIS CPCI.

Status return values shall be used in the S field of the Status ICE as shown in figure 2. The TISERR field shall contain one of the status codes from table VIII. The VMSERR field shall contain the numeric code for the VMS error.

3.1.1.2.1.3 Description of PSS logical I/O interface. This section describes the detailed protocol for all operations across the SSIM. The TIS CPCI communicates across the SSIM with a 16-character string called the Interface Control Element (ICE). ICE is sent from the PSS to initiate all file management operations, all queue operations and operator communications. ICE is sent from the TIS CPCI in response to PSS commands. The TIS CPCI initiates data transfers by interrupting the PSS. The PSS then starts the data transfer across the SSIM.

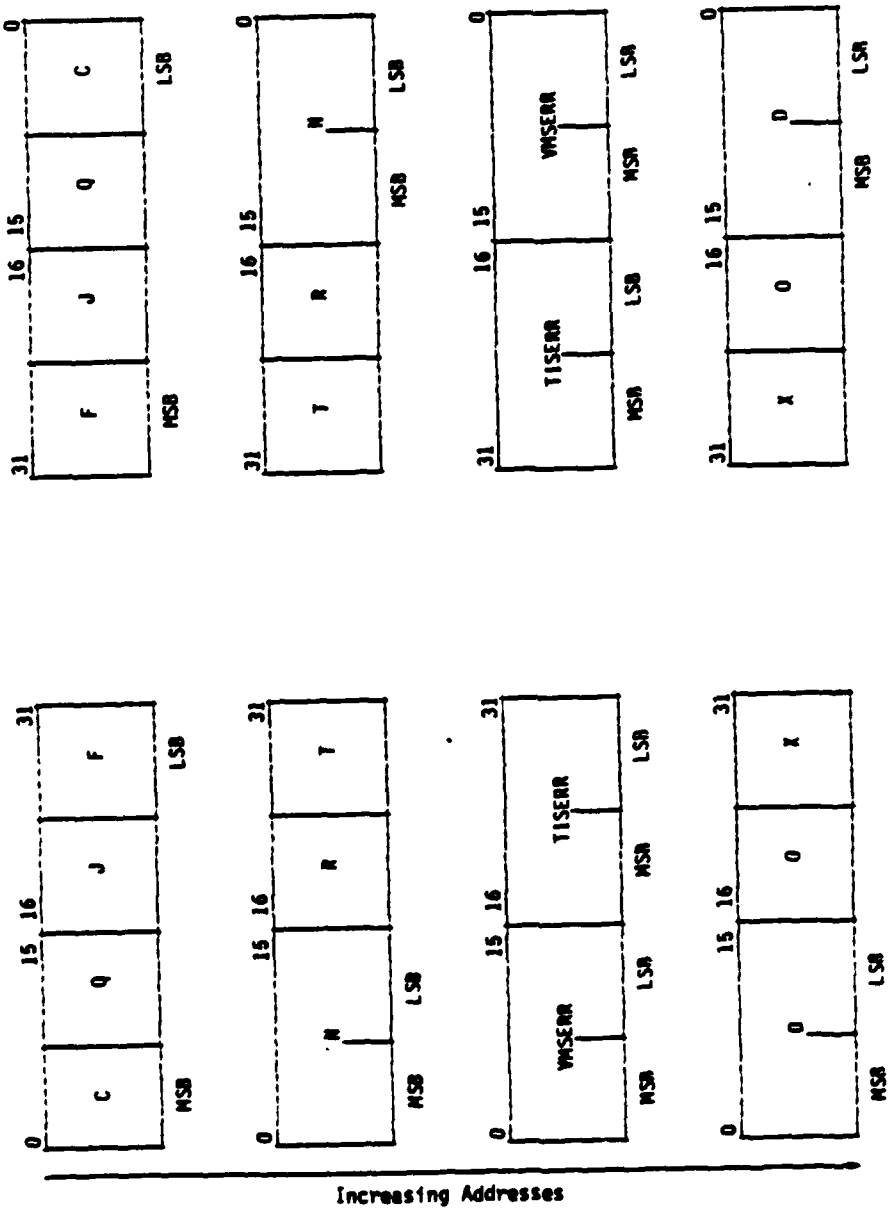
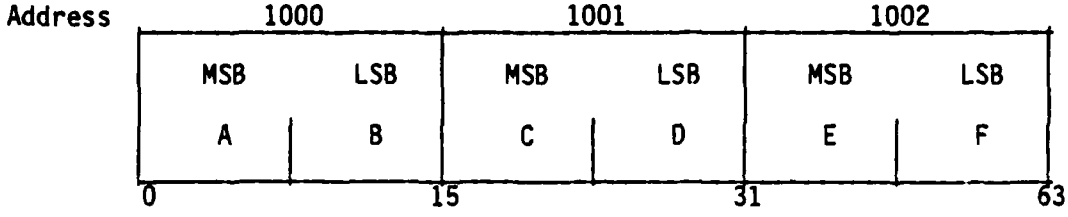
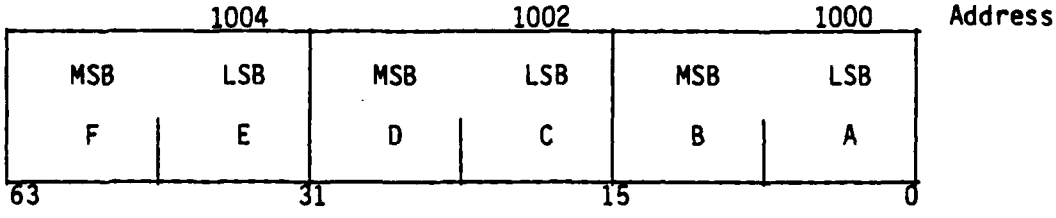



Figure 2. Exchange of ICE Packets by the PSS and TIS




AN/GYK-12 character data buffer
as stored by the PSS prior to output
through the SSIM




SDDS VAX-11/780 character data buffer
as stored by the TIS subsequent to
reading from the SSIM

Figure 3. Exchange of Character Data Buffers
by the PSS and the TIS

## Table IV. Format of ICE Packet

Fixed length: 16 bytes.

| MNEMONIC | SIZE | DESCRIPTION |
|---|---|---|
| C | Byte | Command 1 = Write a file<br>2 = Read from a file<br>3 = M&D (Not processed)<br>4 = Open a file<br>5 = Close a file<br>6 = Delete a file<br>7 = Start input queue, open input and output queue<br>8 = Start input queue, open input queue<br>9 = Start output queue, open output queue<br>10 = Terminate queue<br>11 = Read input queue<br>12 = Write output queue<br>13 = Write message to operator<br>14 = System startup message<br>15 = Read AN/GYK-12(V) system file<br>16 = Test SSIM<br>17 = Error detected<br>18 = Manipulate physical tape<br>19 = Open a punch queue<br>20 = Write a punch queue<br>Command 21 = Manipulate logical disk file |
| Q | Byte | I/O queue table number associated with each request. All communications for a particular operation reference the same table entry. |
| J | Byte | Job slot number. This uniquely identifies the PSS user job requesting the specified operation. |

Table IV. Format of ICE Packet (continued)

| MNEMONIC | SIZE | DESCRIPTION |
|---|---|---|
| F | Byte | File number. (Also device number for tape I/O and tape subcommand). |
| N | Word | Block size in full words [AN/GYK-12(V)]. (Also used for number of files or records to space for disk and tape subcommand). |
| R | Byte | Retry transaction counter. |
| T | Byte | Transaction counter ($0 \leq T \leq 15$). When the most significant bit of the byte is set, then reset counter to indicated number in the low-order 4 bits. |
| S | Long Word | Status information, consisting of two one-word subfields:<br>1. TISERR — Status code from TIS CPCI (see table VIII).<br>2. VMSERR — Status code from VMS. (This value must in some cases conform to the RSX-11D error codes used in the TACFIRE PDP-11 SPS.) |
| D | Word | Direct access record number for disk file I/O |
| O | Byte | Sub-commands for physical tape or logical disk file manipulation. These sub-commands only have meaning when command is 18 or 21.<br>1. Write<br>2. Read<br>3. Forward space N records<br>4. Forward space N files (tape only)<br>5. Back space N records<br>6. Back space N files (tape only)<br>7. Rewind<br>8. Rewind/unload (tape only)<br>9. Write tape mark (tape only)<br>10. Mount tape allocated through JOB card (tape only)<br>11. Dismount tape (tape only)<br>12. Mount tape not previously allocated (tape only)<br>13. Sense write-protect status (tape only) |
| X | Byte | Reserved for future use. |

Table V. Usage of Fields in ICE Packets

| FUNCTION | FIELDS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | Q | J | F | N | R | T | S | D | O | X |
| Open a PSS disk file | 4 | Used | Used | Used | Used | No | No | No | No | No | No |
| Close a PSS disk file | 5 | Used | Used | Used | No | No | No | No | No | No | No |
| Read an open PSS disk file | 2 | Used | Used | Used | Used | No | No | No | Used | No | No |
| Write an open PSS disk file | 1 | Used | Used | Used | Used | No | No | No | Used | No | No |
| Delete a PSS disk file | 6 | Used | Used | No | Used | No | No | No | No | No | No |
| Start job | 7<br>8<br>9 | Used | Used | No | Used | No | No | No | No | No | No |
| Terminate job | 10 | Used | Used | No | No | No | No | No | No | No | No |
| Read from job input queue | 11 | Used | Used | No | No | No | No | No | No | No | No |
| Write to job output queue | 12 | Used | Used | No | Used | No | No | No | No | No | No |

Table V. Usage of Fields in ICE Packets (continued)

| FUNCTION | FIELDS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | Q | J | F | N | R | T | S | D | O | X |
| Write PSS operator message to TIS | 13 | Used | Used | No | Used | No | No | No | No | No | No |
| Unsolicited operator message from TIS | No | Used (DRDB) | No | No | No | No | No | No | No | No | No |
| Error Event | 17 | Used | Used | No | No | No | No | Used | No | No | No |
| Test SSIM Message | 16 | Used | Used | Used | Used | Used | Used | Used | Used | Used | Used |
| System startup | 14 | Used | No | No | Used | No | No | No | No | No | No |
| Write tape Record | 18 | Used | Used | Used | Used | No | No | No | No | 1 | No |
| Read tape Record | 18 | Used | Used | Used | Used | No | No | No | No | 2 | No |
| Forward space n tape records | 18 | Used | Used | Used | Used | No | No | No | No | 3 | No |
| Backspace n tape records | 18 | Used | Used | Used | Used | No | No | No | No | 5 | No |

Table V. Usage of Fields in ICE Packets (Continued)

| FUNCTION | FIELDS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | Q | J | F | N | R | T | S | D | O | X |
| Forward space n tape files | 18 | Used | Used | Used | Used | No | No | No | No | 4 | No |
| Backspace n tape files | 18 | Used | Used | Used | Used | No | No | No | No | 6 | No |
| Rewind tape | 18 | Used | Used | Used | No | No | No | No | No | 7 | No |
| Rewind and Unload tape | 18 | Used | Used | Used | No | No | No | No | No | 8 | No |
| Write EOF tape mark | 18 | Used | Used | Used | No | No | No | No | No | 9 | No |
| Mount previously allocated tape | 18 | Used | Used | Used | No | No | No | No | No | 10 | No |
| Release tape and dismount | 18 | Used | Used | Used | No | No | No | No | No | 11 | No |
| Mount previously unallocated tape | 18 | Used | Used | Used | No | No | No | No | No | 12 | No |
| Sense write protect and tape status | 18 | Used | Used | Used | No | No | No | No | No | 13 | No |

Table V. Usage of Fields in ICE Packets (continued)

| FUNCTION | FIELDS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | Q | J | F | N | R | T | S | D | O | X |
| Start punch output queue | 19 | Used | Used | No | No | No | No | No | No | No | No |
| Write to punch output queue | 20 | Used | Used | No | Used | No | No | No | No | No | No |
| Read System File | 15 | No | No | No | No | No | No | No | Used | No | No |
| Write to open PSS disk file subcommand | 21 | Used | Used | Used | Used | No | No | No | Used | 1 | No |
| Read from open PSS disk file subcommand | 21 | Used | Used | Used | Used | No | No | No | Used | 2 | No |
| Forward space n PSS disk file records | 21 | Used | Used | Used | Used | No | No | No | No | 3 | No |
| Backspace n PSS disk file records | 21 | Used | Used | Used | Used | No | No | No | No | 5 | No |
| Rewind PSS disk file | 21 | Used | Used | Used | No | No | No | No | No | 7 | No |

## Table VI. Open Command ICE Extension

Maximum length: 48 bytes.

| FIELD NAME | SIZE IN BYTES | VALUES | DESCRIPTION | DEFAULT VALUE |
|---|---|---|---|---|
| Filename | 8 | ASCII | PSS filename. | None |
| Member name | 8 | ASCII | PSS member name. Not used if final disposition is temporary. | None |
| Version number | 4 | Binary | If specified, must be in the range of 1 to 32767 decimal. If not specified (zero), the newest version of the specified file will be opened, or the next highest unused version number will be assigned upon creation of a new file. | Newest Version of file |
| Device type | 1 | Binary | Must indicate disk with a value of 7. | None |
| Initial disposition | 1 | Binary | 1 — New file to be created. An attempt by PSS to create a new member in a PSS file dedicated to source-type data will cause an error.<br>2 — Existing file to be accessed. An attempt to access a member in a PSS file dedicated to source-type data with other than block-size equal to an integer multiple of 21 decimal will cause an error.<br>3 — Shared access of existing file. The access parameter must be read only. (Shared implies an existing file.) | New |
| Final disposition | 1 | Binary | 1 — Delete.<br>2 — Keep.<br>3 — Pass (Tape only).<br>4 — Temporary. | Delete |
| Access | 1 | Binary | 1 — Read only.<br>2 — Read and write.<br>3 — Update.<br>4 — Modify.<br><br>Legal pairs of combinations of initial disposition and access are as follows:<br><br>Existing file and read only.<br>Existing file and update.<br>Existing file and modify.<br>Shared file and read only.<br>New file and read and write.<br>New file and update.<br><br>Any other combination of initial disposition and access parameters will cause an error. For all access types other than modify, the record pointer (number) is positioned at the first record of the file; for modify, the record pointer will be positioned at the first free record beyond end-of-file. Modify is treated as an append. | Read and write |

Table VI. Open Command ICE Extension (continued)

| FIELD NAME | SIZE IN BYTES | VALUES | DESCRIPTION | DEFAULT VALUE |
|---|---|---|---|---|
| Block size | 2 | Binary | Size of the largest block in the file in full-words. The N field of the Read File ICE cannot be greater than the block size; however, if the N field in the Read File ICE is less than the block size, then the N field value will be used on a read. This procedure is also followed for a Write File ICE. The block size cannot be greater than 2048. | 128. |
| Space | 2 | Binary | Number of blocks to be allocated initially for file creation. | 20. |
| Extends | 2 | Binary | Number of blocks in each extent. | 20. |
| Record Access | 1 | Binary | 1 — Sequential access only.<br>Any other value — Direct access.<br>The record number (D field in the ICE) may be specified for either Read File ICE or Write File ICE.<br>If the record number (or block number) is not specified in an ICE, then the next sequential record will be retrieved or written; if the record number is present, then the specified record is accessed. | None |
| Classification | 1 | Binary | 1 — CONFIDENTIAL.<br>2 — SECRET. | Unclassified |
| Volume | 2 | ASCII | Disk unit number. Not used by the TIS CPCI. | None |
| Not used | 14 | | Spare bytes. | |

Table VII. Delete Command ICE Extension

Maximum length: 48 bytes.

| FIELD NAME | SIZE IN BYTES | VALUES | DESCRIPTION |
|---|---|---|---|
| Filename | 8 | ASCII | PSS filename. |
| Member name | 8 | ASCII | PSS member name. |
| Version number | 4 | Binary | Must be in the range of 1 to 32767 decimal. |
| Not used | 28 | | Spare bytes. |

Table VIII. Status Return Codes from the TIS CPCI to the PSS

| TIS Return Codes to be Used in S (Status) field of Error ICE | |
|---|---|
| Value | Description |
| 0 | Successful completion |
| 1 | End of volume encountered |
| 2 | End of file encountered on Read |
| 3 | Beginning of file/tape |
| 4 | Not used |
| 5 | Tape write-protected |
| 6 | Invalid access and/or disposition on Open |
| 7 | File name not found in TIS catalog |
| 8 | Member name not found in TIS catalog |
| 9 | Version number not found in TIS catalog |
| 10 | Block size too large on Open |
| 11 | Error detected by VMS in Open processing (Refer to VMS error) |
| 12 | Error detected by VMS in Close processing (Refer to VMS error) |
| 13 | Error detected by VMS in Delete processing (Refer to VMS error) |
| 14 | No job available on Start Input Queue request |
| 15 | Job attempted to open Temp file without previously closing it |
| 16 | Illegal final disposition on Open |
| 17 | Delete attempted on Open file |
| 18 | Close attempted on Closed file |
| 20 | Input Queue has been suspended for PSS |
| 21 | Tape cannot be mounted |
| 22 | TIS catalog cannot be accessed |
| 23 | Illegal volume number for tape request |
| 26 | Tape unit 0 allocated (Mounted) |
| 27 | Tape unit 1 allocated (Mounted) |
| 257 | Illegal ICE command |
| 258 | Illegal job number |
| 259 | Illegal file number |
| 260 | Duplicate file number on Open |
| 261 | Duplicate job number on start job |
| 262 | Duplicate Queue table number for request |
| 263 | Buffer size too large (N field in ICE too large) |
| 264 | I/O error on Disk/Tape read (Refer to VMS error) |
| 265 | I/O error on Disk/Tape write (Refer to VMS error) |
| 268 | Direct access record number too large |
| 276 | Illegal tape command |
| 277 | Tape not reserved for job |
| 278 | Tape not mounted |
| 279 | Illegal tape number |
| 280 | Illegal number of classes (no more than 20 allowed) |
| 281 | File Access Table Full |
| 282 | File deadlock prevention — job cancelled with no dump. |

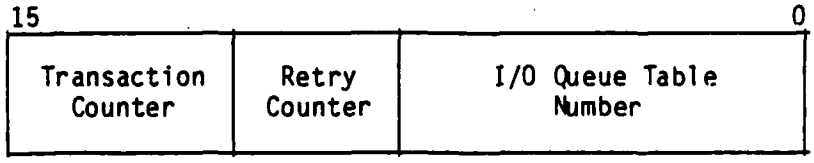
The I/O Queue Table Number, the retry transaction counter, and the transaction counter shall be used in the DRDB as shown in table IX. All data transfers occur from IOU address $16_8$. IOU address $17_8$ is used for TIS CPCI control interrupts only.

3.1.1.2.1.3.1 PSS initiated write ICE string. This procedure allows the user to interrupt the TIS and send an ICE string across the SSIM for processing. All data transfers are sent from IOU address $16_8$. The I/O protocol is as follows:

AN/GYK-12(V) Initiated — Write ICE String

| AN/GYK-12(V) | VAX-11/780 |
|---|---|
| 1. Form the ICE string in a PSS buffer. The T field of the ICE string must contain the next sequential transaction number expected by the TIS. For each ICE request, the transaction number is incremented by one. The transaction number increments from 0 to 15; when 15 is reached, the transaction number is reset to zero. If a retry condition occurs, the T field stays the same but the R field must be incremented by one for each retry attempt. | |
| 2. ITR the SSIM at address $16_8$. If BUSY bit in status byte of the ITR word is set, wait for interrupt. | |
| 3. Initialize keyword and termword for device address $16_8$ to transmit ICE. | |
| 4. Issue DEV =X'4F0E' | |
| | 5. Interrupt received on VAX-11/780. |
| | 6. The CIF and BUSY flags will be set in the DRST. |
| | 7. Initialize DR11-B reigsters: DRWC = 8 words (16 byte ICE string), DRBA = buffer address for ICE, DRST has flags IE and GO set with FNCT = 100, and DRDB = Ø |
| 8. IOU transmits data across SSIM. | |

Table IX. DRDB Register for TIS Initiated Operations




VAX-11/780 Orientation Shown

| Most Significant Byte | Least Significant Byte |
|---|---|

| I/O Queue Table Number Values | Meaning |
|---|---|
| 1-127 | The range of values for the AN/GYK-12(V) PSS I/O queue table numbers. |
| 128-253,0 | Reserved for future growth. |
| 254 | Unsolicited operator message from the TIS. The status byte contains the number of bytes in the operator message. |
| 255 | TIS-initiated ICE string. |
| Retry Transaction Counter | This 4-bit item ranges from 0 to 15. Non-zero values indicate the retry count for the transaction. |
| Transaction Counter | This 4-bit item ranges from 0 to 15. Each VAX-11/780 initiated I/O operation uses this incremented counter. If a transaction is to be retried, the Transaction Counter stays the same but the Retry Transaction Counter is incremented. |

Note: The PSS will receive the value of the DRDB with the bytes exchanged by the SSIM (this procedure is the same for every data transfer); therefore, the PSS will interpret the most significant byte of the DRDB as the I/O Queue Table Number.

10. EOB interrupt.
11. Termword interrupt status byte must show END flag set with ERR flag clear. If an interrupt occurs on address $17_8$ and INT flag is set, the operation in progress on address $16_8$ is cancelled with a device stop command. The interrupt from the VAX-11/780 must be serviced first.

9. ICE transmitted into buffer.
10. Completion interrupt.
11. If no error conditions shown in DRST, keep ICE in VMS buffer until read by Queue-I/O; otherwise, log error with VMS and deallocate buffer.

3.1.1.2.1.3.2 TIS initiated write ICE string. This procedure describes the steps necessary to send an ICE string across the SSIM to the PSS. The ICE is returned when logical or unrecoverable I/O errors occur, or when status must be returned at completion of a logical request; for example, on a PSS open file request, the completion status must be returned to the PSS. The procedure is as follows:

TIS Initiated — Write ICE String

| AN/GYK-12(V) | VAX-11/780 |
|---|---|
| | 1. Test SSIM busy. If busy, wait for interrupt to signal completion of current operation. |
| | 2. Initialize DR11-B registers: DRWC = Queue-I/O word count (must be 16 bytes), DRBA = Queue-I/O buffer address, DRST has flags IE and GO set with FNCT = 010, and DRDB = Queue-I/O parameter from caller (must be 255 decimal). |
| 3. Attention interrupt on IOU address $17_8$. | |
| 4. ITR SSIM. | |
| 5. The DRDB in the ITR word contains an I/O queue table number of 255. | |
| 6. PSS interprets the I/O queue table number as a request to write ICE from the TIS. | |
| 7. The keyword and termword on device address $16_8$ are set up to receive an ICE string. | |
| 8. Issue DEV =X'2COE' which causes SSIM to start the transfer of data into the AN/GYK-12. | |

9. EOB interrupt.
10. If no errors shown in termword interrupt byte, process next request.

9. Completion interrupt.
10. If no errors shown in DRST, process next request; otherwise, log error with VMS and return error status to caller.

3.1.1.2.1.3.3 Send or receive data. The PSS ICE may indicate that additional data is to be transmitted across the SSIM; for example, each read or write of a file requires two I/O operations across the SSIM: the ICE string and one transfer of file data; for every data transfer request from PSS, there will be a corresponding ICE string. The I/O protocol is the same for all PSS initiated operations which require a data block transfer across the SSIM. The procedure is as follows:

Send or Receive Data

AN/GYK-12(V)

VAX-11/780

1. Test SSIM busy. If busy, wait for interrupt to signal completion of current operation.
2. Initialize DR11-B registers: DRWC = Queue-I/O word count (must be same as full-word record size passed in ICE for reading or writing the requested data), DRBA = Queue-I/O buffer address, DRST has flags IE and GO set with FNCT = 010, and DRDB = Queue-I/O parameter from caller (must be I/O queue table number for the ICE initiating the request for reading or writing the record) with Transaction Counter and Retry Counter.

3. Attention interrupt on IOU address $17_8$.
4. ITR the SSIM.
5. INT, ADD, and BUSY flags are set in the status byte of the ITR word.
6. I/O queue table number is passed in the most significant byte of the ITR word. Initialize keyword and termword on address $16_8$ for operation with parameters from the I/O queue table entry corresponding to the I/O queue table number.

7. Issue DEV =X'2COE' or DEV =X'2FOE' for input from SSIM or output to SSIM respectively.
8. EOB interrupt.
9. If no errors, process next request.

8. Completion interrupt.
9. If no errors shown in DRST, process next request; otherwise, log error with VMS and return error status to caller.

3.1.1.2.1.3.4 Bootstrap protocol across the SSIM. The SSIM normally performs I/O operations to address $16_8$ and $17_8$ on the AN/GYK-12(V) IOU. During the bootstrap mode, the SSIM performs the data transfer to address $11_8$ on the AN/GYK-12(V) IOU. The TIS reads the first physical record of the logical bootstrap record into SDSS VAX-11/780 memory. The number of bytes to be transferred during the bootstrap operation are derived from bytes 9, 13, and 25 of the logical bootstrap record. The TIS sets up the DR11-B registers as follows:

| Register | Content |
|---|---|
| DRWC | Word count of the record. |
| DRBA | Memory address of the boot record. |
| DRDB | Zero. |
| DRST | Flags IE and GO set with FNCT = 011. |

If the TIS CPCI does not have enough memory available to transfer the logical bootstrap record in one operation, the logical record will be segmented. The individual segments are sent to the AN/GYK-12(V). After the first segment is sent, the TIS is interrupted by the SSIM. The TIS then reads in the next segment of the bootstrap record and initiates the data transfer across the SSIM. During the bootstrap mode, the SSIM does not send an interrupt to the AN/GYK-12(V).

3.1.1.2.2 VMS interface. The VMS, executing on the SDSS VAX-11/780, provides the user program environment for the TIS CPCI. The basic context of the user environment is the process, which is the entity that the VMS schedules for execution. Each VMS process has an independent address space in which an image executes. Each image executing in a process can call VMS system service procedures to acquire resources and request special processing services from the VMS.

All direct interfaces between the TIS CPCI and the VMS shall be through calls to the VMS system services, which provide all primary resource request activities, such as I/O processing and interprocess communication. Images that use the VMS system services can be written in assembly language or any native-mode programming language that has a CALL (effectively a CALLS or a CALLG facility) statement.

3.1.1.2.3 RMS interface. The VMS provides two request interfaces for performing input and output operations: the I/O system services and the VAX/VMS Record Management Services (VAX-11 RMS). VAX-11 RMS provides a general purpose file and record programming interface that handles most I/O processing. The I/O system services will be used by the TIS CPCI where the storage media is independent of the Files-11 Level 2 and Files-11 Level 1 structures (for example, foreign PSS tapes). The TIS CPCI shall use VAX-11 RMS services for all I/O operations to on-disk files.

VAX-11 RMS provides a set of application-callable procedures that provide facilities for data storage, retrieval, and modifications. The TIS CPCI shall use the VAX-11 RMS sequential, relative, and block I/O file organizations.

3.1.1.2.4 VAX-11/780 input/output subsystem. The following paragraphs discuss the characteristics of the SDSS VAX-11/780 computer on which TIS will execute, and describes the input-output operation of the VAX-11/780 and the computer-peripheral device interfaces.

The VAX-11/780 is a high-speed, general purpose, microprogrammed, microelectronic digital computer. Figure 4 shows the major components of the SDSS VAX-11/780 processor. The principal characteristics of the VAX-11/780 are as follows:

a. Executes variable length instruction operands in native mode and non-privileged PDP-11 instructions in compatibility mode.

b. The processor includes an 8 Kbyte cache, integral memory management, sixteen 32-bit registers, 32 interrupt priority levels, an LSI-11 intelligent console emulator, a programmable real-time clock, and a time-of-day and date-of-year clock.

c. VAX-11/780 native mode instruction set provides 32-bit addressing capability to a maximum of 4 billion bytes of virtual address space. (This maximum is not supported by the VMS operating system.)

d. Memory management hardware includes mapping registers, page protection by access mode (read only, read-write, no access), and hardware translation of virtual to physical addresses.

e. Sixteen 32-bit registers that can be used as storage, accumulators, index, and base registers.

f. Thirteen basic addressing modes.

g. CALL facility completely implemented in hardware with extensive context protection features.

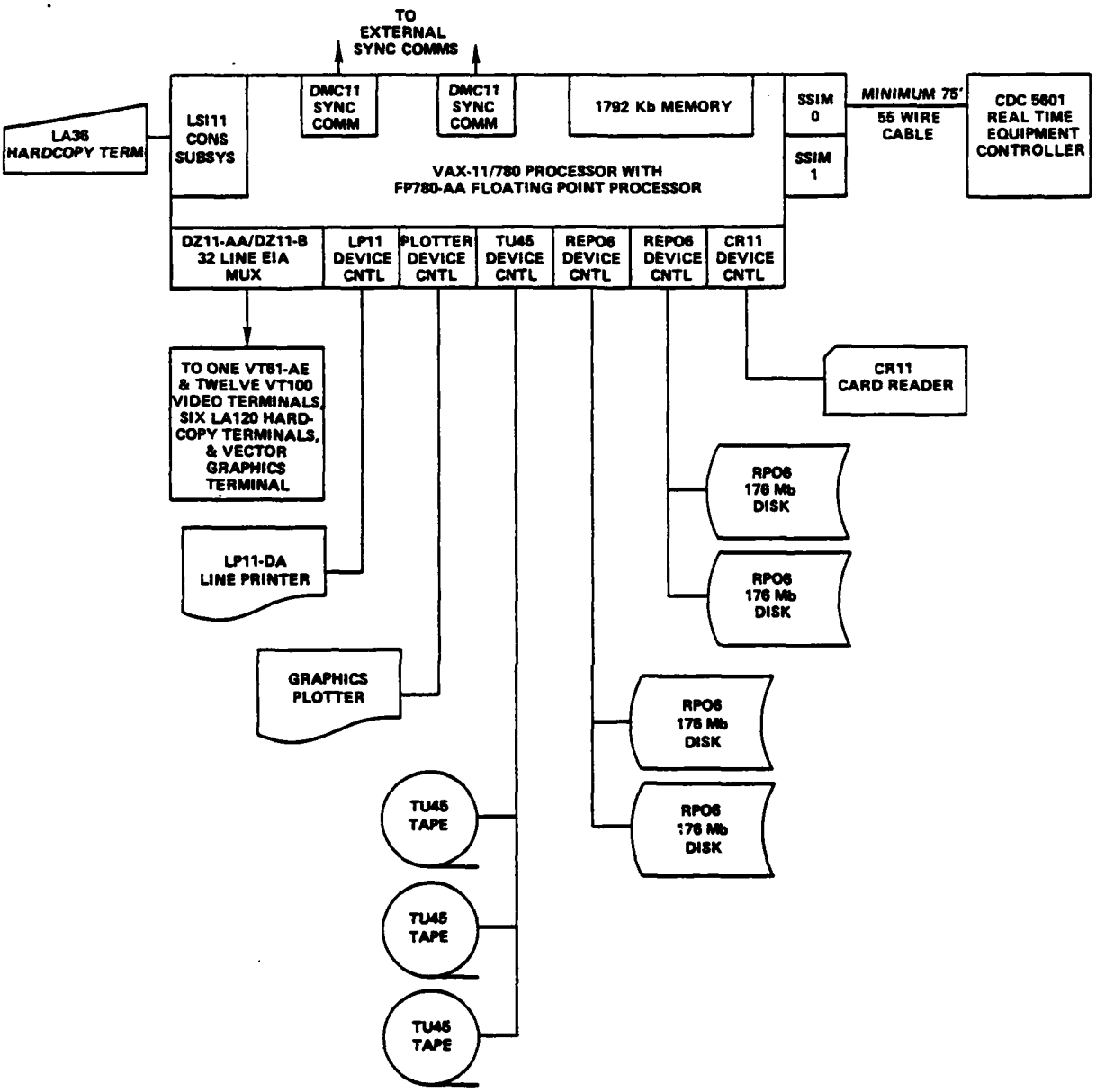



**FIGURE 4. SDSS VAX-11/780 COMPUTER SYSTEM CONFIGURATION**

h. ECC MOS memory expandable to 8 Mbytes.

i. Native mode instruction set includes fully hardware-implemented decimal, character string, and floating point instructions (with Floating Point Accelerator).

j. Peripheral bus adapters allow PDP-11 Unibus and Massbus peripherals to connect to a synchronous 32-bit bus (Synchronous Backplane Interconnect).

k. High-speed (180 µsec) process switching with hardware-implemented context save and restore instructions.

l. I/O throughput rates of 13.3 Mbytes/second theoretical maximum.

The VAX-11/780 computer is well suited to applications that are computationally intensive and to many applications that are I/O intensive. Many real-time applications with very stringent timing requirements (such as signal processing data acquisition and certain communications functions) cannot be satisfied with the VAX-11/780 computer due to the hardware and software overheads introduced into the processing path by the components of the VAX-11/780 Input-Output subsystem.

The VAX-11/780 Input-Output subsystem consists of an internal synchronous backplane interconnect (SBI) bus and adapters to interface the Unibus and Massbus peripherals to the SBI. The Unibus adapter (UBA) interfaces the peripherals on the Unibus to the SBI; the Massbus adapter (MBA) interfaces the peripherals on the Massbus to the SBI. Four separate Unibus adapters and four separate Massbus adapters can be installed on the VAX-11/780. Table X shows the theoretical throughput maximums of each bus structure.

The SBI has a 30-bit address space and can transfer 32 bits or 64 bits of data every cycle. The SBI is capable of an aggregate theoretical throughput rate of 13.3 Mbytes/second. Parity and error checking occurs every 200 nanosecond cycle to provide data integrity. The Unibus has an 18-bit address space and can transfer 8 bits or 16 bits of data every cycle. The Massbus has a 22-bit address space and can transfer 8 bits or 16 bits of data every cycle. The UBA and MBA interfaces perform the following functions:

a. Translate bus addresses to the 30-bit SBI address.

b. Buffer four 16-bit data transfers into a single 64-bit SBI data transfer.

c. Prefetches the next 64-bit quadword from memory before the previous data transfer to a peripheral device has completed.

d. Interrupt and contention arbitration is performed for the peripheral devices local to the adapter.

Table X. Theoretical Maximum Throughput Rates of VAX-11/780 Device Buses

| BUS TYPE | MAXIMUM THEORETICAL THROUGHPUT (Mbyte/sec) |
|---|---|
| SBI | 13.3 |
| UNIBUS | 1.5 |
| MASSBUS | 2.0 |

e. Enables the CPU to directly access peripheral device control and status registers.

f. Translates bus device interrupt requests to the SBI equivalent request.

All peripheral device control and status registers are assigned addresses in physical I/O address space. No special I/O instructions are necessary for ordinary I/O control; values are set and reset in the command and status registers by ordinary processor instructions to command and interrogate a peripheral device.

There are two data transfer modes used in the VAX-11/780: direct memory access (DMA) and programmed I/O. DMA transfers minimize CPU involvement in the I/O activity by transferring large blocks of data to and from memory with the CPU receiving a single interrupt upon completion. Programmed I/O requires that each data element be transferred to and from memory under direct software control and generates an interrupt for each data element. Usually large disk drives are interfaced to the Massbus and use DMA transfer techniques; programmed I/O is required in peripheral devices that transfer comparatively small amounts of data at inherently slow rates such as unit record peripherals. The data transfer between the SSIM and the SDSS VAX-11/780 shall be in DMA mode.

The electrical details of interfacing a DMA or programmed I/O peripheral device to the VAX-11/780 Massbus or Unibus are identical to the interface to the same bus on the PDP-11 computer family.

The major difference between the Unibus and Massbus hardware on the VAX-11/780 and PDP-11 computers is in the interrupt latency, that is, the length of time required between the time that a peripheral device generates an interrupt request and the first instruction of the device interrupt service routine is entered. The interrupt latency for the VAX-11/780 computer is approximately one order of magnitude greater than the corresponding time for many PDP-11 computers. This is the result of additional hardware and software overhead introduced by the UBA and MBA electronics in the VAX-11/780 I/O subsystem. The software overhead for each VAX-11/780 device interrupt is caused by the execution of an interrupt dispatcher routine which receives the interrupt generated by the UBA or MBA, scans an internal CPU register to determine which peripheral device generated the interrupt, and then dispatches the interrupt to the appropriate device interrupt service routine. This hardware and software overhead causes the VAX-11/780 interrupt latency times to range from 10 to 55 microseconds while the corresponding interrupt latency time for a PDP-11/70 computer ranges from 1.5 to 4.0 microseconds.

The increased interrupt latency of the SDSS VAX-11/780 will not significantly decrease the performance of the SDSS TIS software when compared to the TACFIRE PDP-11 SPS/PSS software.

3.1.1.2.5 DR11-B interface. This section specifies the software interface requirements for the DR11-B interface to the SSIM.

JBDRIVER will be a user written device driver for the two DR11-B devices. The JBDRIVER shall be written in accordance with the VAX/VMS Guide to Writing a Device Driver (AA-H499B-TE). The template driver (TDRIVER) supplied with VMS and the DA11-B interprocessor link driver (DADRIVER) source code will be used as guidance in developing the JBDRIVER.

The DR11-B is a general-purpose, DMA interface to the Digital Equipment Corporation (DEC) PDP-11 or VAX-11/780 Unibus. The DR11-B interface contains four software-accessible registers: command and status, word count, bus address, and data. JBDRIVER will inititate an operation by loading the word count register with the two's complement of the number of words to be transferred, specifying the initial memory address where the block transfer is to begin by loading the bus address register, and loading the status and command register with the appropriate function, interrupt enable, and status bits. The GO bit in the status register is then set to initiate the requested DR11-B function.

Figure 5 shows the four DR11-B registers that the JBDRIVER will access to control each DR11-B device and to pass data through each DR11-B to an SSIM.

The word count register (DRWC) is a 16-bit read-write register. JBDRIVER shall load the DRWC register with the two's complement of the number of words to be transferred into or out of the SDSS VAX-11/780 memory. This register will increment towards zero after each word is transferred. When overflow occurs the READY bit of the status and command register will be set. The DRWC is cleared by INIT, the initialization signal produced in the power-up sequence or by the boot switch on the VAX-11/780 console.

The bus address register (DRBA) is a 16-bit read-write register. Bit zero of the DRBA, corresponding to address line A00, will be provided by the SSIM. JBDRIVER shall load the DRBA with the beginning memory address of a buffer for the current data to be transferred. The DRBA Register is incremented by two, by the DR11-B, for each word transferred. If overflow occurs, the ERROR bit in the status and command register is set. This condition is cleared by reloading the DRBA or by the INIT signal. The DRBA is always cleared by the INIT signal.

The status and command register (DRST) is a 16-bit read-write register through which commands are given to the SSIM and the JBDRIVER is provided with the status of the DR11-B and the SSIM. Table XI defines the functions and operations provided by the DRST register.

The data buffer register (DRDB) is a 16-bit read/write register for transferring data between the SSIM, the SDSS VAX-11/780 memory, and the AN/GYK-12 memory.

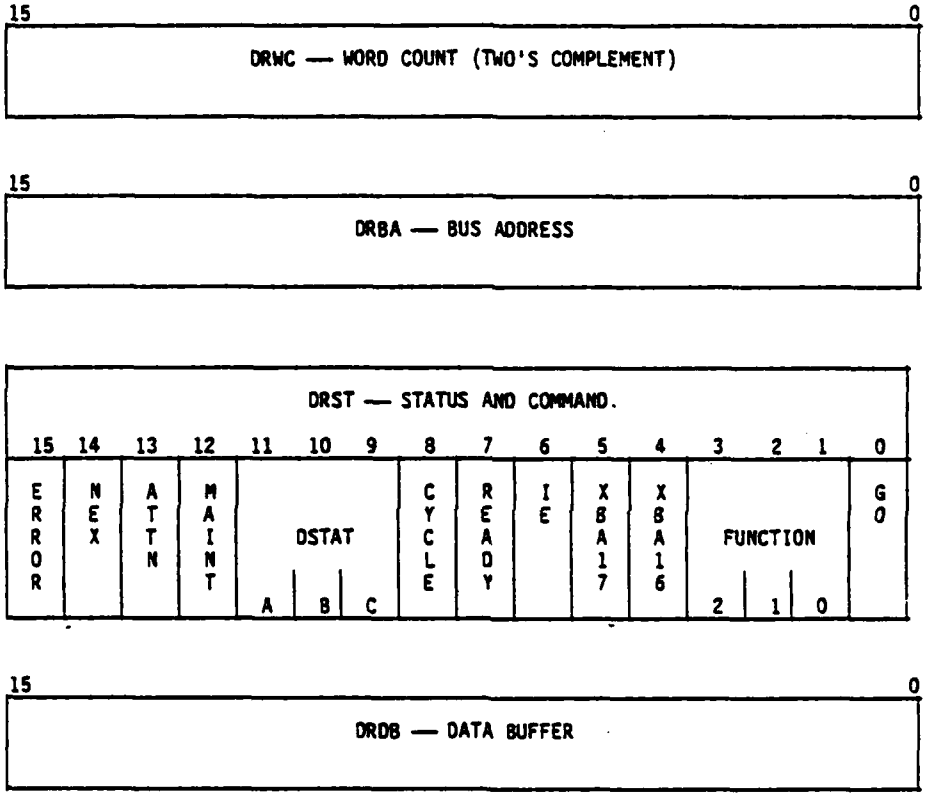



Figure 5. DR11-B Registers

Table XI. DR11-B Command and Status Definitions

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 15 | ERROR (read only) | 1. Indicates one of the following conditions:<br>a. NEX is set.<br>b. ATTN and either PE or CIF is set.<br>c. ATTN is set and both PE and CIF are clear. This indicates that the SSIM has been reset or that the AN/GYK-12 has issued a Device Stop command.<br>d. Bus address overflow (DRBA incremented from all ones to all zeros).<br>2. Causes READY to be set and causes interrupt if IE is set.<br>3. Cleared by removing all three of the following conditions:<br>a. NEX is cleared by loading NEX with a zero or by INIT signal.<br>b. ATTN is cleared by SSIM.<br>c. Bus address overflow cleared by loading DRBA. |
| 14 | NEX | 1. Non-existent memory (NEX) indicates that as the Unibus Master the DR11-B did not receive a SSYN response from memory within 20μs after asserting MSYN.<br>2. Sets ERROR.<br>3. Cleared by INIT signal or loading NEX with a zero. NEX cannot be loaded with a one. |

Table XI. DR11-B Command and Status Definitions (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 13 | ATTN (read only) | 1. Set by CIF, PE, by Device Stop command from AN/GYK-12, or by SSIM Device Reset command from AN/GYK-12.<br>2. Sets ERROR.<br>3. Cleared when READY is cleared. |
| 12 | MAINT | 1. Maintenance bit used in conjunction with the DEC M968 maintenance module to test the DR11-B (*does not* test SSIM).<br>2. Cleared by INIT signal. |
| 11 | CIF (read only) | 1. Control Information Flag (CIF) indicates to the VAX-11/780 software that a request to output an ICE from the PSS is pending.<br>2. Sets ATTN.<br>3. Cleared upon AN/GYK-12 EOB termination when ICE is passed.<br>4. Cleared by SSIM Reset command. |
| 10 | BUSY (read only) | 1. The BUSY flag indicates to the VAX-11/780 that the SSIM is busy with an AN/GYK-12 loop test or data transfer.<br>2. Cleared by AN/GYK-12 EOB termination, by SSIM Reset command from the VAX-11/780, by Device Stop command from AN/GYK-12, or by SSIM Device Reset command from AN/GYK-12. |

Table XI. DR11-B Command and Status Definitions (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 09 | PE (read only) | 1. Parity Error (PE) flag indicates that a parity error was detected on the AN/GYK-12 to SSIM interface during a transfer from AN/GYK-12 to SSIM operation.<br>2. Sets ATTN.<br>3. Reset by SSIM Reset command (see FUNCTION bits) or by PE Reset command. |
| 08 | CYCLE | Not used in the SSIM. |
| 07 | READY (read only) | 1. Indicates that the DR11-B is able to accept a new command.<br>2. Set by INIT signal, ERROR, ATTN, or word count overflow.<br>3. Cleared whenever GO is set.<br>4. Causes interrupt if IE is set. Forces DR11-B to release control of the Unibus and prevents further DMA cycles. |
| 06 | IE | 1. Enables interrupt to occur when either ERROR or READY is set.<br>2. Cleared by INIT signal. |
| 05<br>04 | XBA17<br>XBA16 | 1. Extended bus address bits 17 and 16; in conjunction with DRBA, specifies Unibus address for DMA transfers.<br>2. Cleared by INIT signal. |

Table XI. DR11-B Command and Status Definitions (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 03-01 | FUNCTION | 1. Function field — Three-bit field set by the VAX-11/780 software which defines the command for the SSIM operation; used in conjunction with GO. These bits are defined as follows:<br>a. 001 Interrupt MMP on Channel 6. Loads internal SSIM DRDB register with value currently in DR11-B DRDB.<br>b. 010 Interrupt MMP on Channel 7. Loads internal SSIM DRDB register with value currently in DR11-B DRDB.<br>c. 011 Bootstrap (Channel 11 load) enable.<br>d. 100 Start I/O operation.<br>e. 101 Parity Error Reset.<br>f. 110 Reset SSIM. |
| 00 | GO (write only) | 1. Causes a pulse to be sent to the SSIM to execute the command set in the Function field.<br>2. Clears READY and allows DMA operation. |

Note: All DRST bits are read/write unless otherwise specified.

Figure 6 shows the physical interface of the DR11-B.

Table XII defines the interface signals between the SSIM and the DR11-B.

The SSIM will generate the following interrupts to the VAX-11/780 when the DR11-B driver has enabled interrupts (IE flag is set in DRST):

a. Programmed interrupt from the SSIM, at which time the CIF flag and ATTN flag shall be set.

b. Parity error detected in the SSIM, at which time the PE flag and ATTN flag shall be set.

c. Error detected in the DR11-B interface unit, at which time the ERROR flag shall be set.

d. DR11-B word count register (DRWC) counts to zero at which time the READY flag shall be set.

3.2 Detailed functional requirements. This section specifies, for each external stimulus, the functional flow through the TIS CPCI and the resulting responses. External stimuli originate from the PSS and the TIS Manager and PSS users. TIS shall validate each stimulus, perform the required functions, and return a response. The TIS CPCI shall communicate with the PSS, the TIS Manager and PSS users through the VMS.

In the following subsections each function is specified by defining the inputs to the function, defining the functional processing requirements, and defining the outputs of the function. Section 3.2.14.1, Human Performance, specifies functions that are common to the major functions and specifies the Man Machine Interface (MMI) functional requirements by defining the syntax and grammar of the TCL, and the interaction between the TIS Manager, PSS user, and the TIS CPCI.

Figure 7 and table XIII summarize the overall functional flow of the TIS CPCI. The TIS CPCI will be implemented as twelve major functions. The control flow and data flow specifies the relationship of the major TIS CPCI functions.

JBDRIVER is not shown in the TIS Functional Flow, figure 7, and is considered a part of VMS. When a process issues a Queue I/O to JBDRIVER, the VMS SYS$QIO system service performs functions required by VMS on the request before queuing the request to the JBDRIVER.

The functional flow block diagram shows the internal TIS functional flow, and the VMS and VAX-11 RMS are external to the block diagram. These functions are summarized as follows:

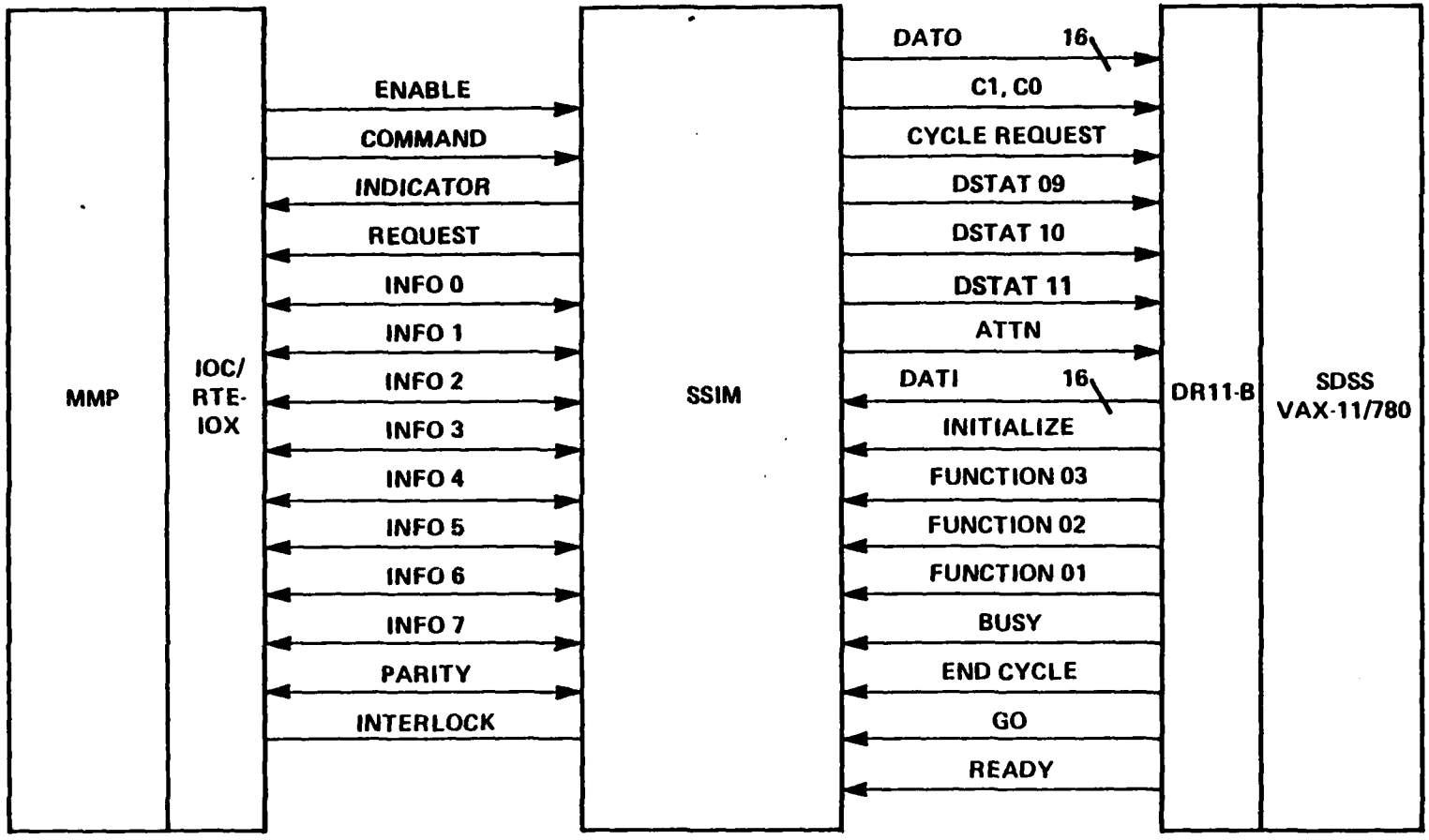



FIGURE 6. PHYSICAL INTERFACES BETWEEN THE DR11-B, SSIM, AND THE RTE IOX CHANNEL

Table XII. Functions of SSIM Signal Lines to the DR11-B Interface

1. DATO 15-00: Sixteen lines used to transfer information from the SSIM to SDSS VAX-11/780 memory.

2. $C_1C_0$: Two lines used by the SSIM to specify that either an input or output transfer is to take place. When $C_0$ is zero and $C_1$ is zero, 16 bits of data will be transferred from the SDSS to the SSIM. When $C_0$ is zero and $C_1$ is one, 16 bits of data will be transferred from the SSIM to the SDSS.

3. Cycle Request: This line will be used by the SSIM to signal the start of a data transfer, either to or from the SDSS.

4. DSTAT 09, 10, 11 Device status bits:

   a. DSTAT 09 (PE) will be set by the SSIM to indicate a parity error has been detected on the MMP interface during a transfer from the MMP to the SSIM.

   b. DSTAT 10 (Busy) is set by the SSIM to indicate to the SDSS that a loop-back test of the MMP interface is in progress or that the SSIM is presently transferring information to or from the MMP.

   c. DSTAT 11 (CIF, Control Information Flag.) This bit will be set by the SSIM when a request to output control information (ICE packets) to the SDSS has been received by the SSIM from the MMP.

5. ATTN: The Attention line is automatically set when the SSIM detects a parity error on the MMP interface or when the SSIM sets the control information flag.

6. DATI 15-00 OUT: Sixteen lines used to transfer information from SDSS VAX-11/780 memory to the SSIM.

7. Initialize: This line, when set by the SDSS, will cause a reset in the SSIM. The SSIM logic will return to the stand-by mode upon receipt of this signal.

8. FNCT 3-2-1: These bits will be used in seven combinations to define a command for the SSIM.

Table XII. Functions of SSIM Signal Lines to the DR11-B Interface (continued)

| | |
|---|---|
| 001<br>010 | Initiate an interrupt sequence to the MMP on channel 16 (001) or channel 17 (010). Loads internal SSIM DRDB register with value currently in DR11-B DRDB. |
| 011 | Enables SSIM to accept commands from channel 11 for bootstrap load. |
| 100 | Start I/O transfer. |
| 101 | Reset Parity Error (PE) flag. |
| 110 | Reset SSIM to stand-by mode. |

9. Busy: The Busy signal, generated by the SDSS, is an indication that the SDSS has received a Cycle Request. The Busy signal will be used to clear the Cycle Request logic in the SSIM.

10. End Cycle: This line is pulsed when the SDSS has completed a cycle transferring information to or from the SSIM.

11. GO: A pulse on the GO line transfers a status, function, or data word into or out of the SDSS.

12. Ready: Ready is used in conjunction with a GO pulse to indicate a valid function command to the SSIM.

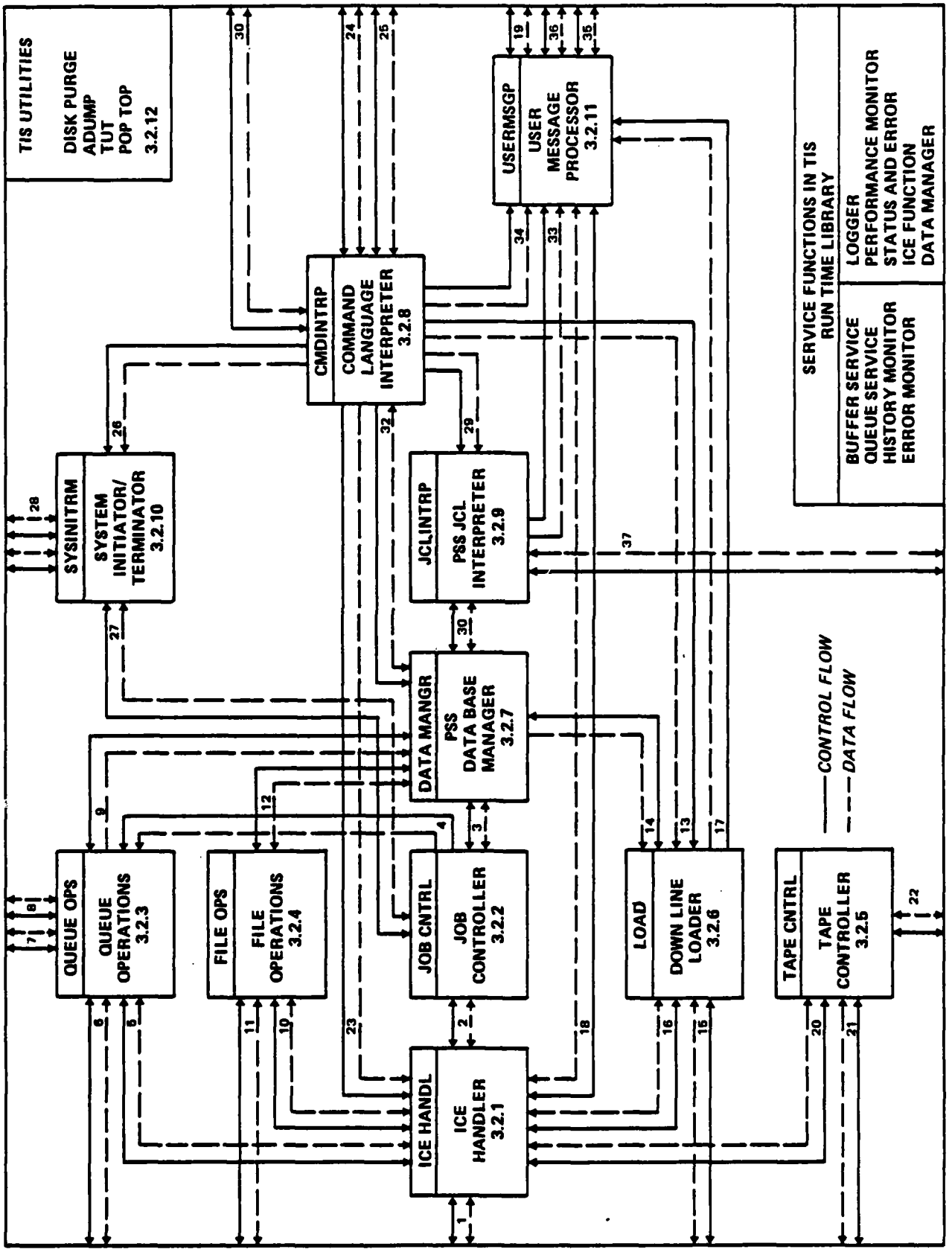



FIGURE 7. TIS FUNCTIONAL FLOW BLOCK DIAGRAM

## Table XIII. Functional Flow Table

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 1 | ICEHANDL | VMS | 1. $QIO with AST completion (JBDRIVER) | 1. QIO parameter list for ICE transfer<br>2. QIO parameter list for ICE extension transfer<br>3. Status ICE or Error ICE to PSS |
| | VMS | ICEHANDL | 1. AST completion | 1. Completion status<br>2. ICE packet |
| 2 | ICEHANDL | JOBCNTRL | $QIO to mailbox | 1. Start input job ICE<br>2. Start print queue ICE<br>3. Open punch queue ICE<br>4. Terminate job ICE |
| | JOBCNTRL | ICEHANDL | Call to run time library | 1. Status ICE parameters<br>2. Error ICE parameters |
| 3 | JOBCNTRL | DATAMANGR | Subroutine call | 1. Close PSS file request (terminate job) |
| | DATAMANGR | JOBCNTRL | Call return | 1. Completion status |
| 4 | JOBCNTRL | QUEUEOPS | $QIO to mailbox | 1. Terminate job ICE |
| 5 | ICEHANDL | QUEUEOPS | QIO to mailbox | 1. Read input job ICE<br>2. Write print queue ICE<br>3. Write punch queue ICE |
| | QUEUEOPS | ICEHANDL | Call to run time library | 1. Status ICE parameters<br>2. Error ICE parameters |

## Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 6 | QUEUEOPS | VMS | $QIO (JBDRIVER) | 1. QIO parameter list to transfer PSS input job<br>2. QIO parameter list to receive print queue and punch queue data |
| | VMS | QUEUEOPS | AST | 1. Completion status<br>2. Print queue data<br>3. Punch queue data |
| 7 | QUEUEOPS | RMS | RMS call | 1. Call parameters to transfer job input from RMS file<br>2. Call parameters to write punch queue data to RMS file |
| | RMS | QUEUEOPS | Call return | 1. Input job<br>2. Completion status |
| 8 | QUEUEOPS | VMS | $SNDSMB (send message to Print Symbiont manager | 1. $SNDSMB parameter list to spool print output from PSS |
| | VMS | QUEUEOPS | Immediate return | 1. Return status |
| | QUEUEOPS | VMS | $QIO (Mailbox Driver) | 1. QIO parameter list to read Print Symbiont completion status |
| | VMS | QUEUEOPS | AST | 1. Print spool completion status |
| 9 | QUEUEOPS | DATAMANGR | Call to run time library | 1. Request to FIND PSS CALL file |
| | DATAMANGR | QUEUEOPS | Call return | 1. Completion status<br>2. Equivalent RMS specification for PSS CALL file |

## Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 10 | ICEHANDLE | FILEOPS | QIO to mailbox | 1. Open PSS file ICE with extension<br>2. Read PSS file ICE<br>3. Write PSS file ICE<br>4. Close PSS file ICE<br>5. Delete PSS file ICE with extension<br>6. Manipulate logical disk file ICE |
| | FILEOPS | ICEHANDL | Call to runtime library | 1. Status ICE parameters<br>2. Error ICE parameters |
| 11 | FILEOPS | VMS | $QIO (JBDRIVER) | 1. QIO parameter list to transfer file data to and from PSS |
| | VMS | FILEOPS | Set event flag with AST completion | 1. Completion status<br>2. PSS file data |
| 12 | FILEOPS | DATAMANGR | Subroutine call | 1. Read PSS file request<br>2. Write PSS file request<br>3. Open PSS file request<br>4. Close PSS file request<br>5. Delete PSS file request |
| 13 | CMDINTRP | LOAD | QIO to mailbox | 1. Boot command |
| 14 | LOAD | DATAMANGR | Subroutine call | 1. Read bootstrap loader<br>2. Read AN/GYK-12 system file |
| | DATAMANGR | LOAD | Call return | 1. Completion status<br>2. Bootstrap loader<br>3. AN/GYK-12 system file |

## Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 15 | LOAD | VMS | $QIO | 1. QIO parameter list to transfer bootstrap loader<br>2. QIO parameter list to transfer AN/GYK-12 system file |
| | VMS | LOAD | AST | 1. Completion status |
| 16 | ICEHANDL | LOAD | QIO to mailbox | 1. Read AN/GYK-12 system file ICE |
| | LOAD | ICEHANDL | Call to run time library | 1. Status ICE parameters<br>2. Error ICE parameters |
| 17 | LOAD | USERMSGP | QIO to mailbox | 1. Boot completion status message to TIS manager |
| 18 | ICEHANDL | USERMSGP | QIO to mailbox | 1. Write PSS message to user ICE |
| | USERMSGP | ICEHANDL | Call to run time library | 1. Status ICE<br>2. Error ICE |
| 19 | USERMSGP | VMS | $QIO (JBDRIVER) | 1. QIO parameter list to input PSS User message |
| | VMS | USERMSGP | AST | 1. Completion status<br>2. PSS user message |
| 20 | ICEHANDL | TAPECNTRL | QIO to mailbox | 1. Manipulate physical tape ICE |
| | TAPECNTRL | ICEHANDL | Call to run time library | 1. Status ICE parameters<br>2. Error ICE parameters |

Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 21 | TAPECNTRL | VMS | $QIO (JBDRIVER) | 1. QIO parameters to transfer tape files or records |
| | VMS | TAPECNTRL | Set event flag | 1. Completion status<br>2. Tape file or record data |
| 22 | TAPECNTRL | VMS | $QIO (magtape driver) | 1. Position tape commands<br>2. Read/write file commands<br>3. Read/write record commands |
| | VMS | TAPECNTRL | AST | 1. Completion status<br>2. File/record data |
| 23 | CMDINTRP | ICEHANDL | QIO to mailbox | 1. Test interface command |
| 24 | CMDINTRP | VMS | $QIO (JBDRIVER) | 1. QIO parameter list to send unsolicited user message to PSS |
| | VMS | CMDINTRP | AST | 1. Completion status |
| 25 | CMDINTRP | VMS | $QIO (terminal driver) | 1. TIS command language prompts<br>2. QIO parameter list to input Manager/user commands |
| 26 | CMDINTRP | SYSINITRM | QIO to mailbox | 1. Clean up command<br>2. Start TIS command |
| 27 | SYSINITRM | JOBCNTRL | QIO to mailbox | 1. Close all files |

Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 28 | SYSINITRM | VMS | Update section system service | 1. Write global common request<br>2. Read global common request |
| | VMS | SYSINITRM | Completion status | 1. Completion status |
| 29 | CMDINTRP | JCLINTRP | QIO to mailbox | 1. Submit command |
| 30 | JCLINTRP | DATAMANGR | Subroutine call | 1. Request to FIND PSS JCL file |
| | DATAMANGR | JCLINTRP | Call return | 1. Completion status<br>2. Equivalent RMS specification for PSS JCL file. |
| 31 | CMDINTRP | RMS | RMS call | 1. Read request for VMS file<br>2. Write request for VMS file |
| | RMS | CMDINTRP | RMS return | 1. Completion status<br>2. VMS file |
| 32 | CMDINTRP | DATAMANGR | Subroutine call | 1. Read PSS catalog files<br>2. Add PSS file name request<br>3. Delete filename request<br>4. Change PSS catalog file entry request |
| | DATAMANGR | CMDINTRP | Call return | 1. Completion status<br>2. PSS catalog file data |

Table XIII. Functional Flow Table (continued)

| LINK | FROM | TO | CONTROL | DATA |
|---|---|---|---|---|
| 33 | JCLINTRP | USERMSGP | Interprocess communication system service | 1. JCL error message request |
| 34 | CMDINTRP | USERMSGP | Interprocess communication system service | 1. TIS Manager/user error message request |
| 35 | USERMSGP | RMS | RMS call | 1. Read error message request |
| | RMS | USERMSGP | Call return | 1. Completion status<br>2. Error message |
| 36 | USERMSGP | VMS | $QIOW (terminal driver) | 1. QIO parameter list to write TIS manager/user message |
| | VMS | USERMSGP | Set event flag | 1. Completion status |
| 37 | JCLINTRP | RMS | RMS call | 1. Read request for JCL file |
| | RMS | JCLINTRP | RMS return | 1. Completion status<br>2. JCL file |
| 38 | SYSINITRM | VMS | QIO (JBDRIVER) | 1. Operator message to PSS to terminate jobs |
| | VMS | SYSINITRM | AST | 1. Completion status. |

The ICE Handler (ICEHANDL) function will perform Queue-I/O directives with AST completion to receive ICE from PSS. The ICE will be validated against the set of legal commands. For the Open File ICE and Delete File ICE, the ICE extensions will be input and appended to the ICE. If any validation errors occur, error or status ICE will be returned to the PSS through a write Queue-I/O to the JBDRIVER. Depending upon the ICE command code, ICEHANDL will transfer the ICE packet to the appropriate TIS process.

ICEHANDL will process the System Startup (class list) ICE command. ICEHANDL will issue a Queue-I/O directive to the JBDRIVER to read the class list initialization data from the PSS and place this data into the TIS parameter tables in the TISGBL common area.

ICEHANDL will send test interface ICE to PSS in one of two modes: periodically or upon command from the TIS Manager. The mode and periodic internal parameters are in the TISGBL area. The ICEHANDL function is specified in 3.2.1.

The Job Control (JOBCNTRL) function will process PSS initiated Start Input Job, Start Print Queue, Terminate Job, and Open Punch Queue command ICE packets. JOBCNTRL will maintain the status of jobs in execution by PSS and awaiting execution by PSS. JOBCNTRL will call the TIS PSS DATAMANGR function subroutines to perform the PSS initiated file functions. The JOBCNTRL function is specified in 3.2.2.

The Queue Operations function (QUEUEOPS) will control the job input, print, and punch queue maintenance and data transfer functions.

The print queue subfunction will process the PSS initiated Write Print Queue ICE packet. QUEUEOPS will issue a Queue-I/O to transfer the print data from the PSS. QUEUEOPS will not directly control the print spooling process since in the VMS operating system this function is normally controlled by the Print Symbiont Manager; instead, QUEUEOPS will interface to the VMS Print Symbiont Manager through a VMS executive directive. Operator control of the print queue will be by the facilities that VMS provides for manipulation of generic and device-specific print queues.

The punch queue subfunction will process the PSS initiated Write Punch Queue ICE command packet. A Queue-I/O directive will be issued to the JBDRIVER to transfer the PSS punch output into the TIS. This data will then be written to disk by VAX-11 RMS. The QUEUEOPS function is specified in 3.2.3.

The File Operations (FILEOPS) function will process PSS initiated Open, Write, Read, Close, Delete, and Manipulate File command ICE packets. For Write and Read requests, FILEOPS will perform a Queue-I/O to the JBDRIVER to transfer the data to the PSS or from the PSS. FILEOPS will process the actual

data transfer to a given disk file through the TIS PSS DATAMANGR function subroutines for files in the PSS database or through VAX-11 RMS for files that are not part of the PSS database. The FILEOPS function is specified in 3.2.4.

The Tape Controller (TAPECNTRL) will process the Manipulate Physical Tape command ICE packet. For Read and Write commands, TAPECNTRL will initiate Queue-I/O directives to the JBDRIVER to transfer the data between the PSS and the TIS. To perform tape control functions, TAPECNTRL will execute Queue-I/O directives to the VMS magnetic tape device driver. TAPECNTRL will execute VMS system services to allocate and logically mount the appropriate PSS tape after the tape has been mounted foreign by the VMS MOUNT command. If an SDSS tape drive is unavailable, then the requesting PSS job will be placed in a wait state and a message periodically displayed to the TIS Manager or to the SDSS VAX-11/780 operator indicating that a free tape drive is needed by the PSS. The TAPECNTRL function is specified in 3.2.5.

The Downline Loader function (LOAD) will receive the Boot command from the the TIS Command Interpreter, read the initial load module from the SDSS VAX-11/780 disk, and issue Queue-I/O directives to the JBDRIVER to load the initial load module into an AN/GYK-12 or the MMP. LOAD will initiate a Queue-I/O directive to the JBDRIVER to transfer the AN/GYK-12 initial load module to the SSIM when the SSIM receives the Channel 11 load command from the AN/GYK-12 IOU. The AN/GYK-12 or MMP will be downline loaded upon actuation of the Channel 11 load momentary pushbutton switch on the IOU maintenance panel, or upon actuation of the equivalent MMP control. LOAD will initiate a Queue-I/O directive to the JBDRIVER to transfer the AN/GYK-12 System File to the PSS when a Read System File ICE packet is received from the PSS. The LOAD function is specified in 3.2.6.

The Data Manager (DATAMANGR) function will be a set of subroutines that provide standardized access and master catalog maintenance functions for the on-disk PSS database. The DATAMANGR subroutines will receive requests from JOBCNTRL, QUEUEOPS, FILEOPS, LOAD, JCLINTRP, SYSINITRM, and CMDINTRP to perform a specific function on a PSS file. DATAMANGR returns to the calling modules completion status and file data for certain functions. The DATAMANGR function is specified in 3.2.7.

The TIS Command Language Interpreter (CMDINTRP) function will issue Queue-I/O directives to the VMS Terminal Driver to prompt the TIS Manager or PSS user, and to input commands. CMDINTRP shall parse the TIS Command Language (TCL) and output error messages through VAX-11 RMS and VMS FORTRAN I/O statements. For the SEND command, CMDINTRP will issue a Queue-I/O to the JBDRIVER to transfer an unsolicited operator message to the PSS. Based on the input command, CMDINTRP will process the command or use the VMS mailbox system service to pass the command to the appropriate TIS CPCI processor. To process the PSS Catalog File maintenance commands, CMDINTRP will call the TIS PSS DATAMANGR subroutines to perform the master user file directory functions.

CMDINTRP will, if commanded by the user, convert PSS formatted files to VAX-11 RMS format files and VAX-11 RMS formatted files to PSS format files. The CMDINTRP function is specified in 3.2.8.

The PSS Job Control Language Interpreter function (JCLINTRP) will scan and validate PSS JCL contained in a PSS file, or a VAX/VMS file. JCLINTRP will be activated by CMDINTRP when the user enters a valid SUBMIT command specifying the file containing the PSS JCL. JCLINTRP will read this file through VAX-11 RMS. JCLINTRP will perform validation and preprocessing functions on the input. Errors will be displayed on the user terminal. If the JCL is valid, and if the user has specified execution, the PSS job is put into the TIS job input queue. The JCLINTRP function is specified in 3.2.9.

The System Initiator/Terminator function (SYSINITRM) will be initiated by the CMDINTRP upon the TIS Manager initiated clean-up CLUP and START commands. SYSINITRM will call the DATAMANGR subroutines to access PSS files for recovery of the PSS database. SYSINITRM will execute a VMS Update Section system service to read and write the TISGBL area. The SYSINITRM function is specified in 3.2.10.

The User Message Processor function (USERMSGP) performs Queue-I/O directives to the VMS Terminal Driver to send messages to the TIS Manager or PSS users. USERMSGP calls VAX-11 RMS to read the message requested by the calling function from the TIS Message File. USERMSGP will receive Write PSS Message to User ICE packets from ICEHANDL. A Queue-I/O to JBDRIVER will input the PSS operator message. The USERMSGP function is specified in 3.2.11.

The TIS Utilities function (TISUTILTY) will be a set of standalone programs, including the TIS and PSS Database Purge function (DISKPURG), the TISGBL Dump function (ADUMP), the Tape Punch function (POPTOP) and the Tape Utility function (TUT). The TISUTILITY function is specified in 3.2.12.

The functions contained in the TISRTL run-time library, which provides services to the TIS processes, will include buffer services, queue services, history monitor, error monitor, and logger. The TISRTL procedures will be reentrant and position independent. The TISRTL functions are specified in 3.2.14.

The following subsections detail the inputs, processing, and output for the major TIS CPCI functions.

3.2.1 Interface control element handler (ICEHANDL). The Interface Control Element Handler performs the functions that are required to handle all ICE packets in a centralized and generic manner.

ICEHANDL interfaces to the VMS system service routine SYS$QIO to send and receive the control data from the JBDRIVER. To receive the ICE packet

generated by PSS, ICEHANDL performs a Queue-I/O with AST completion to the JB devices. This approach is necessary because the ICE packets are issued by the PSS asynchronously; several jobs may be simultaneously executing under each PSS, and each PSS job may have one or more outstanding requests to the TIS. To send ICE packets or to receive the Open File command string or the Delete File command string, which are functionally extensions to the Open File ICE and the Delete File ICE, Queue-I/O operations with AST completion are issued. Receiving ICE extensions or issuing status or error ICE are synchronous events in response to a previously PSS initiated ICE.

For PSS-initiated ICE, ICEHANDL performs control functions, first-level packet validation, and input of ICE extensions when required, and routes the ICE packet to the appropriate TIS functional process as determined by the ICE command field.

For TIS-initiated ICE, Status ICE, or Error ICE, ICEHANDL constructs the ICE packet, and performs control functions.

ICEHANDL interfaces with the TIS functional processes by means of the VMS interprocess communication system services (group-global event flags, the TISGBL, and Mailboxes) to pass the ICE and the ICE extensions. Status and Error ICE will be sent by calls to modules in the TISRTL by the TIS functional process.

ICEHANDL interfaces to the TISGBL to obtain validation data and to write TIS System Startup parameters. ICEHANDL calls the activity monitor routine in the TISRTL to log error and history data.

3.2.1.1 ICEHANDL inputs. ICEHANDL shall receive the following as inputs to the ICE processing functions:

a. ICE packets from the VMS system service SYS$QIO. The format of these packets will be as specified in 3.1.1.2.1.2.

b. Requests from FILEOPS, QUEUEOPS, JOBCNTRL, LOAD, TAPECNTRL, and USERMSGP to send a Status ICE or Error ICE to PSS. The parameter list will be as follows:

Parameter 1: COMMAND Code
Parameter 2: STATUS Code

c. Test command parameters from CMDINTRP.

d. Completion status from the VMS system service SYS$QIO as defined in the VAX/VMS System Services Reference Manual (AA-D018B-TE).

e. System Startup Data from PSS as shown in table V.

3.2.1.2 ICEHANDL Processing.

a. Purpose. The purpose of the ICEHANDL function is to centralize the processing that is common to all incoming and outgoing ICE packets.

b. Approach. ICEHANDL will continually maintain an outstanding read Queue-I/O to the JBDRIVER. ICEHANDL will allocate a buffer from TIS buffer services to receive the ICE packet.

ICEHANDL shall validate the ICE packet fields that are common to all ICE packets. The Command (C) field shall be checked for a value in the range of 1 to 21 decimal. The I/O Queue Table Number (Q) field shall be checked for a value in the range of 1 to 127 decimal. The Job Slot (J) field shall be checked for a value in the range of 1 to 15 decimal. The File Number (F) field shall be checked for a value in the range of 1 to 144 decimal. If any of these validation checks fail, the ICEHANDL shall construct an Error ICE packet and issue a write Queue-I/O to the JBDRIVER to return the Error ICE to the appropriate PSS.

ICEHANDL shall call a VMS interprocess communication system service to pass the ICE to the appropriate TIS functional process. ICE with the C field equal to 1, 2, 4, 5, 6, and 21 decimal shall be passed to File Operations (FILEOPS). ICE with the C field equal to 7, 8, 9, 10, and 19 decimal shall be passed to Job Control (JOBCNTRL). ICE with the C field equal to 11, 12, and 20 decimal shall be passed to Queue Operations (QUEUEOPS). ICE with the C field equal to 13 decimal shall be passed to the User Message Processor (USERMSGP). ICE with the C field equal to 18 decimal shall be passed to the Tape Controller (TAPECNTRL). ICE with the C field equal to 15 decimal shall be passed to the Downline Loader (LOAD).

ICEHANDL shall process Maintenance and Diagnostic (M&D) ICE (C field equal to 3 decimal), System Startup ICE (C field equal to 14 decimal), Test Loop ICE (C field equal to 16 decimal), and Error ICE (C field equal to 17 decimal).

The TIS CPCI will not support the Unit Test SPS (UTSPS) M&D software, which uses the M&D ICE in conjunction with the TACFIRE PDP-11 SPS Maintenance and Diagnostic Reflective Executive (MADRE) CPCEI. If M&D ICE is received, ICEHANDL shall invoke USERMSGP to output a message to the TIS Manager, and shall log the event in the TIS history log.

To process the System Startup ICE ICEHANDLE shall use the size (in words) of the Class List (N field) to obtain a buffer for the Class List and issue a Queue-I/O to JBDRIVER to notify PSS through an interrupt sequence that the Class List is to be sent. ICEHANDL shall write the data to the Class List Table in the TISGBL.

ICEHANDL shall test the SSIM interface to PSS. The TIS Manager will specify, through the TCL SET command, whether the Test Loop ICE packet is to be sent periodically or only upon interactive command, through the TCL TEST command, from the TIS Manager. If the periodic mode is selected, the TIS Manager will specify the time interval. If a periodic test ICE is specified, ICEHANDL will set a time-out timer for the specified time interval. When the timer expires and no data has been successfully transferred in the interval, or when ICEHANDL receives the TCL TEST command from the CMDINTRP, ICEHANDL shall issue the Test Loop ICE and reset the time-out timer. When the Test Loop ICE is received, ICEHANDL shall check bytes 2 through 16 decimal for the binary sequence 0,1,2,...,14 decimal. If the Test ICE sequence is invalid, ICEHANDL shall invoke USERMSGP to send a message to the TIS Manager and shall log the error in the TIS error log.

If ICEHANDL receives Error ICE from PSS, ICEHANDL shall invoke USERMSGP to send a message to the TIS Manager and shall log the error in the TIS error log.

ICEHANDL shall dequeue requests from FILEOPS, QUEUEOPS, JOBCNTRL, LOAD, TAPECNTRL, and USERMSGP to issue Status ICE or Error ICE. ICEHANDL shall build the appropriate ICE packet and issue the Queue-I/O to JBDRIVER.

ICEHANDL shall check the completion status of all VMS System Services issued by the ICEHANDL process.

ICEHANDL shall call the error monitor, when enabled by the TIS Manager, to log all error returns. ICEHANDL shall call the History Monitor, when enabled by the TIS Manager, to log transaction data.

3.2.1.3 ICEHANDL outputs. ICEHANDL shall issue the following outputs:

a. Queue-I/O parameter lists for read and write functions to JBDRIVER.

b. ICE packets with the command field equal to 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 18, 19, 20, and 21 decimal to the appropriate TIS functional processor.

c. Open File command string and Delete File command string ICE extensions to the FILEOPS Process.

d. Messages to the TIS Manager through USERMSGP.

e. Error data to the TIS error log.

f. History data to the TIS history log.

3.2.2 Job control (JOBCNTRL). The Job Control function will initiate and terminate all jobs input to the PSS and the output from the PSS associated

with an input job, or terminate an output job that is not associated with an input job from TIS. This condition may occur for utility functions (for example, FILECOMP) executed independently of the PSS input job queue. JOBCNTRL will receive ICE packets from ICEHANDL, validate the ICE packet, update the TISGBL Job Control Table, and pass to the QUEUEOPS function requests to terminate the output queue for a specified job slot number.

3.2.2.1 JOBCNTRL inputs. JOBCNTRL will receive the following as inputs to this function:

a. A parameter list of fields extracted from a PSS initiated ICE packet. The list is received from ICEHANDL and consists of the following:

Parameter 1. Command Field.
C=7 — Start input job queue.
C=8 — Start output job queue.
C=9 — Start input job queue and output job queue.
C=10 — Terminate job queue.
C=11 — Open punch output queue.
Parameter 2. Job Slot Number.
Parameter 3. Block size of print records or block size of punch records.

b. Completion status from the DATAMANGR subroutines.

3.2.2.2 JOBCNTRL processing.

a. Purpose. The JOBCNTRL function will provide a centralized control mechanism for initiation and termination of PSS input and output jobs, and to centralize the management and control of all data related to the PSS jobs.

b. Approach. JOBCNTRL shall receive, through VMS interprocess communication system services from ICEHANDL, the parameter list as specified in 3.2.2.1.

For the Start Input Job command (C field equal to 7 or 9 decimal), JOBCNTRL shall check the status of the Job Input Queue (JIQ) resident in the TISGBL. If the JIQ has been suspended by the TIS Manager, JOBCNTRL shall call the ICEHANDL status ICE routine in the TISRTL to return Status ICE to the PSS with the status field indicating that the JIQ is currently suspended. JOBCNTRL shall check that the JIQ is not empty and that no other job is executing in this slot. If the JIQ is empty or a duplicate job slot number exists for the specified PSS, Status ICE will be returned indicating these conditions. For a valid request, JOBCNTRL shall assign the job identification number of the job at the top of the JIQ to the requesting PSS job slot, and shall update the Job Control Table (JCT) in the TISGBL.

If the C field is equal to 9 decimal, JOBCNTRL shall check the ICE block size field for a value in the range of 1 to 2048 decimal and, if valid, shall

indicate in the JCT that this job will have associated output. Status ICE indicating successful completion will be returned through ICEHANDL.

For the Start Output Command (C field equal to 8 decimal), JOBCNTRL shall check the JCT to ensure that the job slot number does not exist for another output job from the same PSS. If a duplicate job slot number exists, then Status ICE indicating duplicate job slot number shall be returned. The block size shall be checked for a value in the range of 1 to 2048 decimal and, if valid, JOBCNTRL shall open an entry in the JCT for a valid Start Output Job Request.

For a Terminate Job Command (C field equal to 10 decimal), JOBCNTRL shall access the JCT to verify that the PSS job is active. If not, Status ICE indicating illegal PSS job number shall be returned to PSS. JOBCNTRL shall access the Tape Allocation Table in TISGBL and release any tapes allocated to the terminated job, and the JCT entry for this job shall be deleted. JOBCNTRL shall call the DATAMANGR subroutines to close any files opened by this job. JOBCNTRL shall indicate to QUEUEOPS to dequeue and delete the job from the JIQ, to append the logger file for this job to the Print Queue File for the job, close the file, and enqueue the output print file to the VMS Print Symbiont. If the job has an associated Punch Queue file, JOBCNTRL shall indicate to QUEUEOPS to close the file.

For the Open Punch Queue command (C field is equal to 20 decimal), JOBCNTRL shall check that the job slot does not exist for another punch output job. If valid, a slot in the JCT shall be opened with the job slot number, and the punch output request Status ICE, indicating illegal job slot number or successful completion, shall be returned to the PSS.

3.2.2.3 JOBCNTRL outputs. JOBCNTRL shall produce the following outputs as a result of the processing described in 3.2.2.2.

a. Close PSS file requests to DATAMANGR.

b. Close Print File for job to QUEUEOPS.

c. Request to ICEHANDLE Status ICE routine.

d. Error data to the TIS error log.

e. History data to the TIS history log.

3.2.3 Queue operations (QUEUEOPS). QUEUEOPS will dequeue PSS input jobs from the Job Input Queue and transfer the job statements to the PSS. QUEUEOPS will transfer PSS print and punch output to a VAX-11 RMS file and enqueue the print output, with the logger file appended to the PSS print data, to the VMS Print Symbiont Manager. Punch output can be written to tape by use of the TIS Utility, POPTOP, or by other normal VMS utilities.

3.2.3.1 QUEUEOPS inputs. QUEUEOPS will receive the following as inputs to this function:

a. Terminate job slot request from JOBCNTRL.

b. A parameter list containing fields extracted from PSS initiated ICE packets will be transferred to QUEUEOPS from ICEHANDL. The contents of this list are:

Parameter 1. ICE Command Field (C field).
C=11 — Read job from job input queue.
C=12 — Write print output.
C=20 — Write punch output.
Parameter 2. Job Slot Number.
Parameter 3. Number of words in record for Write Request.

c. Input file of processed PSS JCL for submission to PSS from JCLINTRP.

d. Queue-I/O completion status from JBDRIVER.

e. VAX-11 RMS Completion Status.

f. VMS Print Symbiont Return and Completion Status.

3.2.3.2 QUEUEOPS processing.

a. Purpose. The purpose of the QUEUEOPS function is to control and maintain the Job Input Queue (JIQ) for PSS, the Punch Queue (PUQ) for PSS, and to enqueue print output to the VMS print queue.

b. Approach. QUEUEOPS shall receive, through VMS interprocess communication the parameter list specified in 3.2.1.3 from ICEHANDL.

For a read job from JIQ request QUEUEOPS shall access the Job Control Table (JCT) in the TISGBL and verify that this job slot has an entry and that the JIQ has been opened for this job slot. If no entry exists in the JIQ or if the JIQ has not been opened, Status ICE with an illegal job number indicator shall be returned to PSS by calling the ICEHANDL Status ICE routine in the TISRTL; otherwise, a job shall be dequeued from the JIQ and the file containing the job shall be opened. VAX-11 RMS shall be called to open the file. Records shall be read from the file, blanks appended as necessary to create an 80-byte record, converted from ASCII to EBCDIC with tabs converted to the appropriate number of EBCDIC blanks, assembled into a buffer of 12 card images, and written to the PSS through the SSIM. A Queue-I/O to JBDRIVER shall be issued to transfer the block to the PSS. If the last block does not contain 12 cards, the remainder of the block shall be zero-filled. If a file read request results in an error, QUEUEOPS will call the ICEHANDL Error ICE routine to send the Error ICE packet to the PSS. If the PSS requests a block and end of file is encountered, a Status ICE packet shall be returned to the PSS.

For a Write Print Output request, QUEUEOPS shall issue a Queue-I/O to JBDRIVER to transfer a print record. The word size of the print records shall be obtained from the parameter list. QUEUEOPS shall call VAX-11 RMS to open a file to receive the print output. When the JBDRIVER Queue-I/O completes, trailing blanks shall be stripped from the record, and the print record shall be written into the file by calling VAX-11 RMS. QUEUEOPS, when the terminate job slot request is received by interprocess communication from JOBCNTRL, shall call VAX-11 RMS to read the logger file for a particular job slot. The logger file shall be appended to the Print Output and the file shall be closed. The file shall then be added to the VMS Print Queue by the VMS Send Message to Symbiont Manager ($SNDSMB) system service. When the Symbiont Manager indicates successful completion, the print output file will be deleted by the VMS Symbiont Manager.

If the SUBMIT option field indicates that the job output is to be held for scrolling, then QUEUEOPS will open the print output file in the user account that submitted the original PSS job request.

For the Write Punch Output request, QUEUEOPS shall call VAX-11 RMS to open a file to store the output. A Queue-I/O to JBDRIVER shall be issued to transfer a Punch Output record from the PSS, and the record size will be taken from the ICEHANDL parameter list. When the Queue-I/O completes, the record shall be written to disk by calling VAX-11 RMS. When the terminate job slot request is received from JOBCNTRL, the file shall be closed by calling VAX-11 RMS. If VAX-11 RMS returns a error on a write request, Error ICE shall be sent to the PSS through the ICEHANDL Error ICE routine.

3.2.3.3 QUEUEOPS outputs. QUEUEOPS will produce the following outputs as a result of its processing:

a. Queue-I/O directives to the JBDRIVER to transfer PSS data.

b. VAX-11 RMS calls to open, read, write, close, and delete job input, print output, and punch output files.

c. Calls to the DATAMANGR subroutines to obtain the equivalent VAX-11 RMS file specification for a PSS CALL file.

d. Calls to the VMS Symbiont Manager to enqueue print files to VMS.

e. Error data to the TIS error log.

f. History data to the TIS history log.

3.2.4 File operations (FILEOPS). The File Operations function will process the PSS ICE packets that request file access and manipulation. FILEOPS will receive the ICE parameters from ICEHANDL, validate the request, call the DATAMANGR subroutines, and perform Queue-I/O directives to JBDRIVER to transfer the file data to and from the PSS. FILEOPS will control the number of simultaneously open files.

3.2.4.1 FILEOPS inputs. FILEOPS will receive the following as inputs to this function:

a. A parameter list of fields extracted from a PSS-initiated ICE packet. The list is received from ICEHANDL and consists of the following:

Parameter 1. Command Field.
C=1 — Write file.
C=2 — Read file.
C=4 — Open file.
C=5 — Close file.
C=6 — Delete file.
C=21 — Space file forward.
C=22 — Space file backward.
C=23 — Rewind file.
Parameter 2. Job Slot Number.
Parameter 3. File Number.
Parameter 4. Record size for C=1 and C=2. Number of records for C=21 and C=22.
Parameter 5. Direct Access Record Number for C=2 and C=1. If this parameter is absent, access shall be in sequential mode.
Parameter 6. PSS File Specification for C=4 or C=6. (File name, member name, version number).
Parameter 7. Device Type for C=4.
Parameter 8. Initial Disposition for C=4.
Parameter 9. Final Disposition for C=4.
Parameter 10. Access for C=4.
Parameter 11. Block Size for C=4.
Parameter 12. Initial Allocation for C=4.
Parameter 13. File Extension for C=4.
Parameter 14. Classification for C=4.
Parameter 15. Disk unit number for C=4.

The values and size of these parameters will be as specified in table VI.

b. File data and status information from the PSS Data Manager.

c. File data from the PSS.

d. Completion Status from the JBDRIVER.

3.2.4.2 FILEOPS processing.

a. Purpose. The FILEOPS function will provide all file handling functions requested by the PSS and control the number of files opened such that use of VMS resources does not cause serious degradation of the SDSS.

b. Approach. FILEOPS shall receive, through VMS interprocess communication system services from ICEHANDL, the parameter list specified in 3.2.1.3.

For the Open File command, FILEOPS shall validate that the job slot is marked active in the Job Control Table contained in the TISGBL. FILEOPS shall check that the file number is in the range of 1 to 144 decimal and that the file has not been already opened by the PSS. The Device Type shall be checked for a value of 7 decimal. The Initial Disposition shall be checked for a value in the range of 1 to 3 decimal. The Final Disposition shall be checked for a value of 1, 2 or 4 decimal. The Access parameter shall be checked for a value in the range of 1 to 4 decimal. The Initial Allocation parameter shall be checked for a value in the range of 1 to 100 decimal. The disk unit number parameter will not be used.

The Write File command and Read file command shall be validated by checking that the Job Slot Number is marked active in the JCT. The file shall be checked to verify that the PSS has opened the file in a previous request. The Record Size shall be checked for a value in the range of 1 to 2048 decimal.

The Close File command shall be validated by checking that the Job Slot Number is marked active in the JCT. The file shall be checked to verify that the PSS has opened the file and that the file has not already been closed.

The Delete File command shall be validated by checking that the Job Slot is active and that the file exists and is currently closed.

The Space File commands and Rewind File command shall be validated by checking that the Job is active, that the file exists, and that the file is currently open.

If any validation errors are detected, FILEOPS shall call the Status ICE routine in the TISRTL to return the Status ICE to PSS.

To perform the Write File command, FILEOPS shall issue a Queue-I/O to the JBDRIVER to transfer the data to be written to the file from PSS. Error ICE shall be written if an I/O error occurs; otherwise, if JBDRIVER indicates successful completion, FILEOPS shall call the DATAMANGR Write File subroutine to write the data. If the DATAMANGR Write File subroutine indicates successful

completion, FILEOPS shall call the ICE Status routine in the TISRTL to return Status ICE to the PSS indicating successful completion. If an I/O error occurs, the Status ICE for a disk I/O error shall be sent to the PSS.

To perform the Read command, FILEOPS shall call the Read File DATAMANGR subroutine to read the data from the SDSS disk. A Queue-I/O shall be issued to the JBDRIVER to transfer the data to the PSS. If an End of File is encountered or if an I/O error occurs, the appropriate Status ICE or Error ICE shall be sent to the PSS.

FILEOPS shall handle PSS source files in the following manner. If the file type in the PSSMUFD indicates a 72 byte or 80 byte format librarian (SPSLIB) source file, then FILEOPS shall pad blanks to column 80, while any tabs present shall be converted to the appropriate number of blanks. If the source file is not a library file, then blanks shall be padded to column 80 and tabs corrected to the appropriate number of blanks. All members of PSS source libraries input to FILEOPS will be ASCII encoded and will not be converted to EBCDIC by FILEOPS since the PSS requires these files to be ASCII encoded.

The Close, Delete, Space, and Rewind commands shall be processed by calling the DATAMANGR subroutines. When the operation is performed Status ICE shall be returned to the PSS.

FILEOPS shall allow up to 72 decimal separate PSS files to be opened at one time. If this threshold is exceeded, FILEOPS shall extend the queue and data storage to support n additional open files. The value of n is specified by the TIS Manager with the TCL SET Command. If this value is exceeded, FILEOPS shall throttle subsequent PSS file open requests by queueing up those requests and by marking the JCT to indicate that additional PSS jobs cannot be started. When a sufficient number of active jobs complete, FILEOPS shall unlock the job queue. FILEOPS shall call the Error and History Monitor to log errors and specific transactions.

3.2.4.3 FILEOPS outputs. FILEOPS shall produce the following outputs as a result of the processing functions:

a. Queue-I/O parameter list to JBDRIVER.

b. Call parameter list to DATAMANGR.

c. Call parameter list to Status and Error ICE routines in the TISRTL.

d. Error data to the TIS error log.

e. History data to the TIS history log.

3.2.5 Tape controller (TAPECNTRL). The Tape Controller function will receive Manipulate Physical Tape ICE from ICEHANDL. TAPECNTRL will validate the ICE, allocate the tape resource with prevention of deadlocking, perform the indicated PSS function, and return Status and Error ICE.

3.2.5.1 TAPECNTRL inputs. TAPECNTRL will receive the following as inputs to this function:

a. Parameter list of fields extracted from the Manipulate Physical Tape ICE from ICEHANDL. The contents of the parameter list are:

Parameter 1. Tape Command.

| Value | Function |
|---|---|
| 1 | Write |
| 2 | Read |
| 3 | Space Forward N Records |
| 4 | Space Forward N Files |
| 5 | Space Back N Records |
| 6 | Space Back N Files |
| 7 | Rewind |
| 8 | Rewind/Unload |
| 9 | Write Tape (EOF mark) |
| 10 | Mount Tape (Allocated on Job Card) |
| 11 | Dismount Tape |
| 12 | Mount Tape Not Allocated |
| 13 | Sense Write-Protect Status |

Parameter 2. AN/GYK-12 System identifier.
Paremeter 3. Job Slot Number.
Parameter 4. Tape Unit Number (26 or 27).
Parameter 5. N — Block Size (words) for Tape Commands 1 and 2.
— Number of Records or Files to Space for Tape Commands 3, 4, 5, and 6.

b. Completion status from VMS Magnetic Tape Driver (TMDRIVER).

c. Completion status from JBDRIVER.

3.2.5.2 TAPECNTRL processing.

a. Purpose. The purpose of TAPECNTRL is to handle all PSS tape operations and to control allocation of the tape resources between two Programming Support Systems executing on different AN/GYK-12 computers and between PSS jobs executing on the same AN/GYK-12.

b. Approach. TAPECNTRL shall receive (by interprocess communication from ICEHANDL) and process the parameter list specified in 3.2.5.1.

ICEHANDL shall validate that the value of the tape command is in the range of 1 to 13 decimal and that the Tape Unit Number is 26 or 27 decimal. The PSS tape unit number will be mapped to the appropriate SDSS tape unit number. For Tape Commands 1 and 2, TAPECNTRL shall validate that the block size is in the range of 1 to 2048 decimal. For Tape Commands 3, 4, 5, and 6, TAPECNTRL shall check for a positive value. If any of the validation checks fail, TAPECNTRL shall call the ICEHANDL routine Status ICE in the TISRTL to send the appropriate Status ICE to the PSS.

TAPECNTRL shall issue a QUEUE-I/O to JBDRIVER to transfer tape data. The size of the transfer is determined by the ICE N field. For a Read Tape Command, TAPECNTRL shall issue a read logical block Queue-I/O to the Magnetic Tape Driver to transfer the data to a buffer. Upon successful completion, TAPECNTRL shall issue a Write Queue-I/O to the JBDRIVER. If the tape read is unsuccessful, the ICEHANDL Error ICE routine in the TISRTL shall be called to return Error ICE for an unreceivable I/O error the PSS. For a Write Tape Command, TAPECNTRL shall issue a Queue-I/O to the JBDRIVER to read the data into a buffer and then shall issue a Write Logical Block Queue-I/O to the Magnetic Tape Driver. If the Magnetic Tape Driver returns an error status upon completion, then TAPECNTRL shall call the Error ICE routine in the TISRTL to send Error ICE to the PSS.

Table XIV specifies the type of Queue-I/O for commands 3, 4, 5, 6, 7, 8, 9, and 13 decimal that TAPECNTRL shall issue to the Magnetic Tape Driver. Upon successful completion, except for Rewind and Rewind/Unload, TAPECNTRL shall call the Status ICE routine to return Status ICE to the PSS. If the Queue-I/O completion status indicates an I/O error, the Error ICE routine shall be called to return Error ICE to the PSS.

For the Mount Tape (allocated on job card) and Mount Tape (not allocated on job card) Commands, TAPECNTRL will perform $ALLOC and $ASSIGN system services. The tape must have been previously mounted by the VMS MOUNT command. For the Dismount Tape, TAPECNTRL shall perform a $DEALLOC System Service; then the tape can be dismounted by the VMS DISMOUNT command. TAPECNTRL shall call the Status ICE routine to return Status ICE to the PSS.

3.2.5.3 TAPECNTRL outputs. The following will be output by TAPECNTRL as a result of the tape processing functions:

a. Call list to ICEHANDL Status and Error ICE routines.

b. Queue-I/O parameter list for JBDRIVER to transfer tape records.

c. Queue-I/O parameter list for Magnetic Tape Driver.

d. Error data to the TIS error log.

e. History data to the TIS history log.

Table XIV. TAPECNTRL Queue — I/O Functions

| TAPE COMMAND | QUEUE — I/O FUNCTION |
|---|---|
| 3. Space forward n records | Skip record |
| 4. Space forward n files | Skip file |
| 5. Space back n records | Skip record |
| 6. Space back n files | Skip file |
| 7. Rewind | Rewind |
| 8. Rewind/unload | Rewind offline |
| 9. Write tape mark | Write end of file |
| 10. Sense write protect status | Sense tape mode |

3.2.6 AN/GYK-12 downline loader (LOAD). The AN/GYK-12 Downline Loader (LOAD) function shall be initiated by the TCL BOOT command by the TIS Manager. The BOOT command specifies the AN/GYK-12 computer to be loaded and the PSS file which contains the load module. The PSS system file can reside either on the SDSS VAX-11/780 disk or the SDSS VAX-11/780 tape. This will enable the TIS CPCI to downline load any properly formatted AN/GYK-12 System File or boot tape. LOAD will read the first boot record, determine the length of the initial bootstrap load module, and transfer the record to the AN/GYK-12. When the record has been sent, the next record of the initial load module is read and transferred to the AN/GYK-12. When the entire initial load module has been sent, a timer is started. If the timer expires before a Read System File ICE is received, the System File is closed and a message sent to the TIS Manager.

LOAD validates the Read System File ICE, and reads and transfers the number of records as specified in the Read System File ICE. When the LOAD operation is complete a message is sent to the TIS Manager.

3.2.6.1 LOAD inputs. LOAD shall receive the following inputs:

a. The TCL command BOOT, modified by the AN/GYK-12 SSIM channel number, and the name of the System File to load. The System File name may be either a PSS filename, a VAX-11 RMS filename, or a boot tape which has been mounted as foreign by the VMS. The boot tape may be an LSS-created load module.

b. The AN/GYK-12 System File containing the AN/GYK-12 load module. The first 28 bytes of the load module file will be as specified in table XV.

c. Queue-I/O completion status.

d. Completion status from the TIS data management process for accesses to the PSS database.

3.2.6.2 LOAD processing.

a. Purpose. The purpose of the LOAD function is to downline load, upon command from the TIS Manager, the specified AN/GYK-12 with an AN/GYK-12 load module.

b. Approach. LOAD shall receive notification from CMDINTRP to accept a valid TCL BOOT command from the TIS Manager.

If the specified load filename is a PSS file, LOAD shall initiate the downline load function by an open request to the DATAMANGR subroutines, and a read request to the DATAMANGR subroutines to read the first physical record of the file specified by the BOOT command.

LOAD shall determine if the AN/GYK-12 load module file is valid by checking bytes 0 and 1 for the AN/GYK-12 IOC Automatic Diagnostic Check Byte and Reflected Check Byte, which are $27_{16}$ and $D0_{16}$ respectively. The total

TABLE XV. INITIAL LOAD MODULE FORMAT

| | | BYTE |
|---|---|---|
| 0 | $57_8$ | 0 |
| 0 | 0 | 2 |
| 0 | $320_8$ | 4 |
| 0 | 0 | 6 |
| 0 | Keyword MSB | 8 |
| 0 | 0 | 10 |
| 0 | Keyword Next MSB | 12 |
| 0 | 0 | 14 |
| 0 | Keyword Next LSB | 16 |
| 0 | 0 | 18 |
| 0 | Keyword LSB | 20 |
| 0 | 0 | 22 |
| 0 | Block Length Extension \| 0 | 24 |
| 0 | 0 | 26 |

length of the initial load module shall be determined from the 11 most significant bits of the Keyword and the Block Length Extension Bits.

LOAD shall issue a Queue-I/O to JBDRIVER with a parameter indicating that the current request is a downline load transfer. Upon successful completion of the Queue-I/O, LOAD shall read the next record of the initial load module and issue a Queue-I/O to transfer the record through the SSIM. When the entire initial load module has been transferred, the last record number shall be saved and the AN/GYK-12 System File closed. LOAD shall send a message to the TIS Manager that the initial load module has been successfully loaded. If any I/O errors occur during the bootstrap load process, LOAD shall send an error message to the TIS Manager and shall log the error into the TIS error log.

LOAD shall receive, by interprocess communication from ICEHANDL, the Read System File ICE (C field equal to 15 decimal). Load shall validate the ICE by checking that the value in the Direct Access Record Number (D) field is in the range of 0 to 32767 decimal. If the Direct Access Record Number is zero, the Direct Access Record Number shall be defaulted to 1. LOAD shall then open the System File by a request to the DATAMANGR subroutines, VAX-11 RMS, or by VMS system services for foreign tape. LOAD shall read the next record after the initial load module and shall issue a Queue-I/O to JBDRIVER to transfer the record. LOAD shall repeat this process until the number of records requested in the D field have been transferred or end of file (EOF) is encountered in the AN/GYK-12 System File. When EOF is encountered, LOAD shall call the ICEHANDL Status ICE routine in the TISRTL to send a Status ICE packet indicating EOF on the System File to the requesting AN/GYK-12.

When the file has been transferred, LOAD shall request the DATAMANGR, VAX-11 RMS, or by a VMS system service to close the AN/GYK-12 System File and a completion message shall be sent to the TIS Manager.

LOAD shall call the Error Monitor to log any errors and the History Monitor to log transaction data.

3.2.6.3 LOAD outputs. The following shall be output from the LOAD function:

a. Queue-I/O parameter list to JBDRIVER to transfer the initial load module.

b. Queue-I/O parameter list to JBDRIVER to transfer System File records.

c. Open, Read, and Close requests to the DATAMANGR subroutines to access the PSS database.

d. Open, Read, and Close requests to VAX-11 RMS to access the AN/GYK-12 System File.

e. Read and Write Queue-I/O requests to the VMS to access the AN/GYK-12 System File on foreign tape.

f. Messages to the TIS Manager through USERMSGP.

g. Error data to the TIS error log.

h. History data to the TIS history log.

3.2.7 Data manager (DATAMANGR). The Data Manager functions provide a standardized and centralized access mechanism to the PSS database. The Data Manager function is a collection of subroutines called by other TIS application functions to perform a logical operation on a PSS data file. Since VAX-11 RMS is non-reentrant, the DATAMANGR will consist of a set of subroutines contained in an object module library.

3.2.7.1 DATAMANGR inputs. DATAMANGR will receive the following items as input to this function:

a. Parameter list passed on to DATAMANGR from the calling function. The contents of this list are as follows:

Parameter 1. Function.
1. Open file.
2. Read file.
3. Write file.
4. Close file.
5. Delete file.
6. Find PSS file name in PSS Master User File Directory (PSSMUFD).
7. Find PSS file name and member name in PSSMUFD.
8. Find PSS dataset in PSSMUFD.
9. Space forward.
10. Space backward.
11. Rewind.

Parameter 2. File Identification
Parameter 3. AN/GYK-12 System Identifier.
Parameter 4. Job Identification.
Parameter 5. Status Return Address.
Parameter 6. Address of Parameter List for specific functions.

3.2.7.2 DATAMANGR processing.

a. Purpose. DATAMANGR shall provide standardized maintenance and single server control of the PSS database.

b. Approach. Access to the PSS database will be through the PSS Master User File Directory (PSSMUFD) as specified in Section 3.3. The PSSMUFD will contain an entry for each PSS Directory. Each entry will contain the VAX-11 RMS Directory Name, Device and VMS Directory containing the member names, the default blocksize and file type. The DATAMANGR subroutines shall process all requests to access PSS source and non-source files by the com-

ponents of the TIS CPCI, and will have exclusive access and control of the PSSMUFD.

DATAMANGR shall process calls from FILEOPS to perform the following functions: find a PSS file, open a PSS file, write a PSS file, read a PSS file, close a PSS file, delete a PSS file and to manipulate file requests of space forward, space backwards, and rewind file. DATAMANGR shall call VAX-11 RMS to perform the actual file I/O operation. When an Open File request is received by the DATAMANGR, the parameters in the PSS Open Command String shall be stored in the File Attributes Table. If the PSS Initial Disposition indicates that the file is a new file, then DATAMANGR shall call VAX-11 RMS to create the file using the PSS Block Size to allocate the proper number of virtual blocks. All files will not be kept open in the VAX-11 RMS context but will be opened and closed by the calling TIS CPCI process, through a call to the DATAMANGR open function, when a read or write request is received. If the PSS Initial Disposition indicates a shared file, then DATAMANGR shall control contention for the file. DATAMANGR shall also control contention between requests from two AN/GYK-12 systems for the same file.

DATAMANGR shall process requests to find a PSS Directory Filename, Member Name, and Version Number and open the corresponding VAX-11 RMS file. DATAMANGR shall search the PSSMUFD to find the entry for the Directory Filename. The Device Name and VMS Directory Name shall be extracted. The VMS Device Name comprises the first four characters of the VAX-11 RMS File Specification (for example DBA0). The VMS Directory shall then be read to get the UIC or directory string for the member name and version number. This shall be appended to the VMS Device Name. For example, the PSS file specification DEVSRC,TEST,12 might be converted to the VAX-11 RMS file specification DBA0:[300,310]TEST.DAT;12. All PSS Member Names shall have .DAT appended to them, except that JCL files, where specified by a PSS dataset identification, may have a VAX-11 RMS .JCL file type.

DATAMANGR shall return status information for all I/O requests. DATAMANGR shall call the Error Monitor and History Monitor routines to log errors and certain transactions.

3.2.7.3 DATAMANGR outputs. DATAMANGR will produce the following outputs as a result of the processing:

a. Call parameter list to VAX-11 RMS to find open, read, write, close, and delete files.

b. File data and status information to calling function.

c. Error data to the TIS error log.

d. History data to the TIS history log.

3.2.8 Command interpreter (CMDINTRP). The Command Interpreter will input and parse the TIS Command Language (TCL) specified in 3.2.14.1. Based on the command, CMDINTRP will invoke the appropriate TIS function to process the command. For some commands, CMDINTRP will process the function locally. CMDINTRP provides a first-level prompting capability.

3.2.8.1 CMDINTRP inputs. CMDINTRP will receive the following as inputs to this function:

a. TIS commands as specified in 3.2.14.1. The commands are received from Terminal Driver.

b. Status and HELP text from VAX-11 RMS.

c. Completion Status from JBDRIVER.

3.2.8.2 CMDINTRP processing. CMDINTRP shall read commands entered by a user from a terminal or from VMS batch. CMDINTRP shall check that the command is a valid TCL verb and, if modified by a noun, that the noun is valid for that verb. CMDINTRP shall check that the user has the proper authority to enter the command. CMDINTRP shall provide a HELP capability. If the user types HELP, without any qualifier, then a list of all commands shall be written. If the user types HELP and a TCL verb, a list of nouns and other parameters for that verb shall be written. The HELP text shall be read from the TIS HELP file by calling VAX-11 RMS.

CMDINTRP shall process the SEND command by checking the command syntax and issuing a Queue-I/O directive to JBDRIVER to send the user message to the PSS. When JBDRIVER indicates successful completion, then CMDINTRP shall send a message to the user through USERMSGP that the message was sent to PSS.

The SUBMIT command shall be validated and sent to JCLINTRP for processing.

The CANCEL command shall be validated and the Job Control Table updated to cancel the job.

The CONVERT command shall be validated and processed by CMDINTRP. For CONVERT/PSS, CMDINTRP shall call VAX-11 RMS to input the editible VMS file, convert the file to PSS format and call the DATAMANGR subroutines to write the resultant PSS-formatted file to disk. For CONVERT/VMS, CMDINTRP shall call the DATAMANGR subroutines to input the PSS file, convert the PSS-formatted file to VAX-11 RMS editible format, and write the resultant VAX-11 RMS-formatted file to disk.

SET/QUEUE shall be validated by CMDINTRP. The JIQ shall be locked, and the queue maintenance function shall be performed on the JIQ, and subsequently the JIQ shall be unlocked.

CMDINTRP shall validate the SHOW command, access the TISGBL to obtain the data to display to the user and pass this data to USERMSGP to write to the user.

CLUP and START shall be validated and passed to SYSINITRM by VMS interprocess communication for processing.

TEST shall be validated and passed to ICEHANDL by VMS interprocess communication for processing.

BOOT shall be validated and passed to LOAD by VMS interprocess communication for processing.

CMDINTRP shall validate SET and access the TISGBL to set the parameters in the JCT and the TIS System Parameter Table.

DISPLAY shall be validated and the PSSMUFD shall be accessed through DATAMANGR for display and access by the user.

CREATE, REMOVE, and UPDATE shall be validated and the DATAMANGR routines in the TISRTL will be called, if necessary, to perform the function.

For all validation errors, CMDINTRP shall indicate to the user the field in error.

3.2.8.3 CMDINTRP outputs. CMDINTRP shall produce the following outputs as a result of the processing:

a. Queue-I/O parameter list to JBDRIVER to send operator message to the PSS.

b. Help and Prompt text to terminal driver.

c. User displays to VAX-11 RMS and VMS FORTRAN I/O subsystem.

d. TIS commands to processing modules.

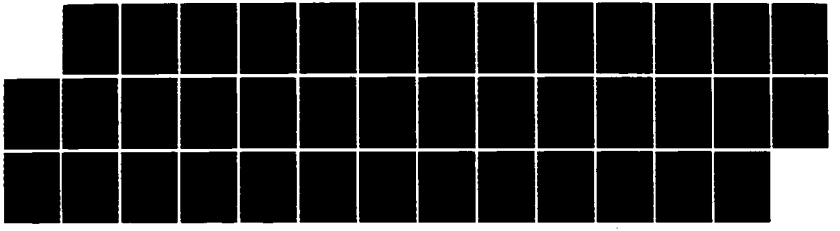
e. Error data to the TIS error log.

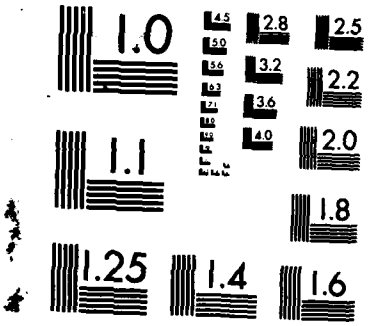
f. History data to the TIS history log.

3.2.9 JCL interpreter (JCLINTRP). The JCL Interpreter function accepts and validates the PSS Job Control Language. The PSS user that initiated the JCL is notified of any errors detected. The job can be held in a file or placed in the Job Input Queue.

AD-A173 427 COMPUTER PROGRAM DEVELOPMENT SPECIFICATION FOR TACTICAL INTERFACE SYSTEM(U) ANALYTICS INC MCLEAN VA 31 JUL 81 1585-TR-02 DAAK80-81-C-0010 2/2

UNCLASSIFIED F/G 9/2 NL






MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

3.2.9.1 JCLINTRP inputs. JCLINTRP will receive the following as input to this function:

a. The SUBMIT command from CMDINTRP.

b. A PSS or VAX-11 RMS file specification for the data file containing the JCL. All data files input to JCLINTRP shall be ASCII encoded.

3.2.9.2 JCLINTRP processing. The JCLINTRP shall accept a PSS file specification or a VAX-11 RMS file specification from the SUBMIT command. JCLINTRP shall call the DATAMANGR subroutines to read a PSS file and VAX-11 RMS to read a VAX-11 RMS file.

JCLINTRP shall process PSS CALL files by calling the DATAMANGR subroutines to obtain the equivalent VAX-11 RMS file specification, and by calling VAX-11 RMS to open, read, and close the input CALL file. JCLINTRP shall process JCL substitution characters if these exist in the input JCL.

JCLINTRP shall perform the following validation. The first card shall be checked for a PSS Job card. If not, all cards to the next Job card shall be flushed and a warning issued to the PSS user. The Job card shall be checked for a PSS Job Identification and a valid class field for the target system. The Job Identification shall be checked that the Job Identification does not duplicate another job currently in the TIS. Any error shall cause the JCLINTRP to flush all cards to the next Job card and to issue an error message to the user. If the tape parameter is specified on the PSS Job card, the value shall be checked for 1 or 2. If not, the Job shall be flushed and an error message output to the user. If the copy parameter is specified, the value shall be checked for the range of 1 to 9 decimal. If not, the Job shall be flushed and an error message output to the user. If the Job card is valid, any forms control information shall be saved in the TISGBL if the job is to be immediately put on the job queue and executed by a PSS. JCLINTRP shall validate embedded JCL. If a ()JCL card is encountered, the characters in columns 7 and 8 are saved. All cards up to a ()FIN card shall be checked, if the characters in columns 1 and 2 match the saved characters, and if a match is found, the characters () shall then be inserted into columns 1 and 2. The card images are scanned until the next ()END card. If no ()END card is found, an error message shall be output and the job shall not be queued.

JCLINTRP shall process the CALL card. When a CALL card is encountered, JCLINTRP shall call the DATAMANGR subroutines or VAX-11 RMS (depending on the CALL file specifier) to read cards from the file specified on the CALL card until a RETN card is encountered.

If the SUBMIT command specified immediate enqueuing for execution by a PSS, then JCLINTRP shall create and fill an output file for each PSS job read from the input PSS JCL. This file shall be ASCII formatted, variable length

records containing the processed PSS JCL to be sent to the requesting PSS through QUEUEOPS. These output JCL files shall be written to the SDSS disk under a directory name dedicated to the TIS job input files.

If the SUBMIT command specified immediate enqueueing for execution by a PSS, then the Queue Service Routine in the TISRTL shall be called to place the equivalent VAX-11 RMS file specification on the Job Input Queue.

3.2.9.3 JCLINTRP outputs. The following items are output by JCLINTRP as a result of the processing:

a. VAX-11 RMS file specification to be enqueued in the Job Input Queue.

b. Output file of processed PSS JCL for submission to PSS through QUEUEOPS.

c. Error Messages to a PSS user.

d. Error data to the TIS error log.

e. History data to the TIS history log.

3.2.10 System Initiator/Terminator (SYSINITRM). The SYSINITRM function will restore the TIS after a PSS crash to the state prior to the crash. If the VMS crashes, the normal VMS backup procedures followed in the SDSS facility will be used to restore the disk.

3.2.10.1 SYSINITRM inputs. The following will be received by SYSINITRM as inputs to this function:

a. The TCL START command.

b. The TCL CLUP command.

3.2.10.2 SYSINITRM processing. SYSINITRM shall periodically check flags in the TISGBL that indicate a change has been made to a critical parameter and therefore that the area must be written to disk to record the change. Depending on the size of the TISGBL, the global area will be partitioned so that the entire contents do not have to be written for every instance of change. There will be a flag for each partition, with each partition defined as zeroable or non-zeroable.

SYSINITRM shall write the PSS Master User File Directory (PSSMUFD) to a disk file whenever an entry in the PSSMUFD is changed. If an error is encountered on this I/O operation, a message shall be sent to the TIS Manager that a backup file has been created for the PSSMUFD. SYSINITRM shall then write the contents of the PSSMUFD to the backup file.

When SYSINITRM receives the START command from the CMDINTRP, SYSINITRM shall check for a warm or cold start. For a cold start, SYSINITRM shall zero all zeroable partitions and write these to disk by calling the VMS Update Section system service. For a warm start, SYSINITRM shall call VMS to read the TISGBL from the disk to the global area. If an I/O error is encountered, or if recovery is not possible, then an error message shall be sent to the TIS Manager indicating that the warm start failed. When the TISGBL is loaded, SYSINITRM shall verify that all files required by jobs in the Job Input Queue still exist by calling VAX-11 RMS to determine file status.

When SYSINITRM receives the CLUP command from CMDINTRP, SYSINITRM shall perform the following:

a. If the appropriate PSS is still active, then SYSINITRM shall send operator messages to the PSS to terminate the current PSS jobs without a dump.

b. If the appropriate PSS is not still active or if no response is received from the terminate messages, SYSINITRM shall send a parameter list to JOBCNTRL indicating that the jobs executing on the specified PSS are to be terminated.

SYSINITRM shall insert a message into each existing job output print file indicating that the job was terminated by the TIS CLUP command.

SYSINITRM shall save the JIQ and restore the JIQ upon a warm start.

3.2.10.3 SYSINITRM outputs. SYSINITRM shall produce the following outputs as a result of the processing:

a. Call parameter list to VAX-11 RMS to write the PSSMUFD to disk.

b. Message to TIS Manager that a backup file has been created for the PSSMUFD.

c. Call parameter list to VMS system service to read TISGBL from disk.

d. Call parameter list to VMS Update Section system service to write TISGBL to disk.

e. Parameter list to JBDRIVER.

f. Queue-I/O parameter list to JBDRIVER.

g. Call parameter list to the DATAMANGR subroutines to determine file status.

h. Error data to the TIS error log.

i. History data to the TIS history log.

3.2.11 User message processor (USERMSGP). The User Message Processor function will output messages to a TIS Manager or a PSS User logged onto the system. The messages will originate from the TIS and from the PSS.

3.2.11.1 USERMSGP inputs. USERMSGP will receive the following as input to this function:

a. A parameter list of fields extracted from the Write message to Manager ICE packet. This input is originated by ICEHANDL and will consist of the following:

Parameter 1. Command Code.
C=13
Parameter 2. Job Slot Number.
Parameter 3. Number of Bytes in Message.

b. A parameter list originated by TIS modules to write a TIS message to a TIS Manager or a PSS User. The parameter list will contain the following:

Parameter 1. TIS Manager/PSS User Identification.
Parameter 2. TIS Message Number.
Parameter 3. Number of Values (n).
Parameter 4. Value 1.
Parameter i+3. Value i.
Parameter i+4. Value i+1.
Parameter n. Value n.

c. Queue-I/O Completion Status from JBDRIVER.

d. Queue-I/O Completion Status from Terminal Driver.

e. Status and Message Text from VAX-11 RMS.

3.2.11.2 USERMSGP processing.

a. Purpose. USERMSGP will provide control of the output interface to a TIS Manager and PSS Users.

b. Approach. USERMSGP shall receive through a VMS interprocess system service a parameter list from ICEHANDL to write a PSS message to a user. USERMSGP shall obtain a buffer to accommodate the number of bytes specified in parameter 3. USERMSGP shall issue a Queue-I/O to JBDRIVER to input the message from PSS. When JBDRIVER indicates successful completion, USERMSGP shall determine which terminal to write the message to using the job slot number. USERMSGP shall call the TIS Logger to log all messages, including those messages originated by the TIS CPCI, into the job output file for the appropriate PSS job. USERMSGP will issue a Queue-I/O to Terminal Driver to

write the message. If Terminal Driver indicates unsuccessful completion, USERMSGP shall call the Error Monitor to log the error.

The parameter list for TIS Messages will be queued FIFO to USERMSGP. USERMSGP will dequeue the request, and call VAX-11 RMS to read the message template from the TIS Message File. USERMSGP shall get the values passed and fill in the template, determine if the user is logged on and to which terminal. USERMSGP shall issue a Queue-I/O to Terminal Driver to write the message. If the message is for a TIS Manager and no TIS Manager is logged on, USERMSGP shall send the message to the SDSS Master Operator Console (OPA0:). If a PSS user is not logged on, the message will be logged.

3.2.11.3 USERMSGP outputs. USERMSGP will produce the following outputs as a result of the processing functions:

a. Queue-I/O parameter list to JBDRIVER to read message from PSS.

b. Call parameter list to VAX-11 RMS to read TIS Message File.

c. Queue-I/O parameter list to Terminal Driver to write a message.

d. Error data to the TIS error log.

e. History data to the TIS history log.

3.2.12 TIS utilities. The TIS Utilities will be four stand-alone functions that are initiated by the Digital Command Language (DCL) RUN command separately from the TCL. These utilities consist of a Disk Purge function (DISKPURG), a TISGBL Dump function (ADUMP), a Tape Utilities function (TUT), and a Print or Punch Tape Output function (POPTOP).

3.2.12.1 TIS utilities inputs. The following will be inputs for the utilities function:

DISKPURG will accept the disk drive number to be purged, from the user.

The TISGBL will be input to ADUMP.

The Tape Unit and a command will be input to TUT by the user. The commands are:

(1) Scratch Tape
(2) Tape Label ANSI
(3) Tape Label EBCDIC
(4) Rewind
(5) Rewind and Unload
(6) Write Tape Mark

(7) Skip Files
(8) Skip Records

The inputs for POPTOP are tape unit number and a command to write the print queue or punch queue or both to tape.

3.2.12.2 TIS utilities processing.

a. Purpose. TIS Utilities will provide disk purge, tape operations and a dump of TISGBL.

b. Approach. DISKPURG shall accept a disk drive number as input. DISKPURG shall write to the first sector of a track with the track number, read the sector back and verify that the same data was read back as was written. DISKPURG shall write a set of random numbers on each track three times and verify that the track contains the third set of random numbers. DISKPURG shall test for bad tracks and output to the user the bad track numbers.

ADUMP shall print an articulated dump of the TISGBL.

TUT shall provide tape labeling and positioning.

POPTOP shall write the print queue or punch queue to tape.

3.2.12.3 TIS utilities outputs. The TIS Utilities will produce the following outputs:

DISKPURG will output bad track numbers.

ADUMP will output a formatted, articulated listing of TISGBL.

TUT will output a tape label and tape positioning commands.

POPTOP will output a print queue or punch queue.

3.2.13 JBDRIVER DR11-B device driver. The JBDRIVER device driver will be a set of routines and tables that the VMS will use to process a device-dependent I/O request. The JBDRIVER shall control the operation of one or two DEC DR11-B direct memory access interfaces to perform I/O between the SDSS VAX-11/780 applications programs and the SSIM for each AN/GYK-12 IOU or the MMP IOU emulation. Reconfiguration of the JBDRIVER to support additional DR11-B to SSIM interfaces will be accomplished through reassembly of the JBDRIVER source. The TIS CPCI will reference the DR11-B to SSIM interfaces serviced by the JBDRIVER as logical devices JBA0 and JBA1.

3.2.13.1 JBDRIVER inputs. JBDRIVER will receive the following as inputs to the I/O processing:

a. I/O request packet from VMS for user I/O preprocessing, start I/O, and I/O postprocessing.

b. Control blocks in the VMS I/O database.

c. A buffer from VMS if the transfer is an ICE packet.

d. A buffer address from the calling user process if the transfer is not an ICE packet.

e. Values in the JBDRIVER driver tables.

f. Queue-I/O function codes from calling user processes as follows:

(1) Read data from JB device, with parameters for buffer address, length in bytes, and value to be placed in the DRDB. The length must be an integer multiple of 4. The buffer must be word aligned.

(2) Write data to JB device, with parameters for buffer address, length in bytes, and value to be placed in the DRDB. The length must be an integer multiple of 4. The buffer must be word aligned.

(3) Read ICE packet from JB device, with parameters for buffer address and length. The length must be 16 bytes, and the buffer may be byte aligned.

(4) Write ICE packet to JB device, with parameters for buffer address and length. The length must be 16 bytes, and the buffer may be byte aligned.

(5) Write PSS Operator Message to JB device, with parameters for buffer address and length in bytes. The length must be an integer of 4, and the buffer must be word aligned.

(6) Channel 11 load (bootstrap) JB device, with parameters for buffer address and length. The length must be an integer multiple of 4. The buffer must be word aligned.

(7) Sense SSIM device characteristics. This function returns the current device-dependent SSIM characteristics in the second word of the I/O status block.

(8) Set SSIM device characteristics. This function allows the caller to change the current device-dependent SSIM operating parameters. The word-aligned address of a quadword characteristics buffer is a required parameter.

(9) M&D read with channel 16 interrupt to the specified JB device, with the same parameters and restrictions as function code (1).

(10) M&D read with channel 17 interrupt to the specified JB device, with the same parameters and restrictions as function code (1).

(11) M&D write with channel 16 interrupt to the specified JB device, with the same parameters and restrictions as function code (2).

(12) M&D write with channel 17 interrupt to the specified JB device, with the same parameters and restrictions as function code (2).

(13) M&D write with channel 11 bootstrap enable to the specified JB device, with the same parameters and restrictions as function code (6).

3.2.13.2 JBDRIVER processing.

a. Purpose. The purpose of the JBDRIVER function is to transfer ICE packets and other data buffers between the PSS and the TIS CPCI through the DR11-B interface to the SSIM, to provide the necessary driver tables to the VMS to describe the DR11-B devices, and to allow the TIS CPCI to bootstrap downline load an AN/GYK-12 or MMP through an SSIM.

b. Approach. JBDRIVER will receive an I/O Request Packet (IRP) from the VMS, validate the parameters, and, if the requested JB device is not busy, initiate the operation. For the transfer of ICE packets, JBDRIVER will allocate a buffer, using standard VMS executive services, from the VMS non-paged dynamic memory. JBDRIVER will use the VMS I/O postprocessing routine to return ICE to the calling process, and the VMS I/O preprocessing routine to obtain ICE from the buffer of the calling process and to move the ICE into VMS non-paged dynamic memory. JBDRIVER shall maintain the time-order of arrival of the ICE from the PSS so that the applications software receives the ICE in the same time-order as the ICE was sent by the PSS. JBDRIVER shall maintain the time-order of TIS-initiated ICE sent to the PSS. Using buffered I/O with a direct data path causes an addition of approximately 19 μsec processing time due to the UBA hardware.

JBDRIVER will use standard VMS executive services to lock larger user buffers in memory, allocate a buffered data path, and to DMA blocks of data between the SDSS memory and the SSIM. The JBDRIVER shall interpret the DRST according to the bits defined in 3.1.1.2.4 and table X. The JBDRIVER shall perform the following functions:

a. Define the DR11-B devices for the rest of the VMS through the driver prologue table.

b. Define the JBDRIVER for the VMS procedure that loads the driver into the VMS virtual address space and that creates the JBDRIVER data structures. This will be accomplished through the driver prologue table.

c. Ready the DR11-B devices for operation at VMS start-up and during recovery from a power failure.

d. Perform device-dependent I/O preprocessing.

e. Translate programmed requests for I/O operations into commands specific to the DR11-B.

f. Activate the DR11-B.

g. Respond to hardware interrupts generated by the DR11-B devices.

h. Respond to DR11-B timeout conditions.

i. Respond to requests to cancel I/O operations on the DR11-B devices. This function is initiated by the VMS $CANCEL system service.

j. Report DR11-B errors to a VMS error logging program.

k. Return status from the DR11-B devices to the process that requested the I/O operation.

l. Return ICE packets to the requesting process, and accept ICE packets from the requesting process.

m. Return device status information to requesting applications processes through the VMS Get Device and Get Channel Information system services.

n. Return device characteristics information to requesting processes.

o. Allow requesting processes to change device-dependent characteristics.

The JBDRIVER will perform I/O transfers in two modes:

a. Buffered I/O for the 16-byte ICE packets. Buffered I/O allows data to be buffered in VMS address space. When the transfer is complete, the data is transferred to the calling user process. JBDRIVER will use a direct data path for small DMA transfers.

b. Direct I/O for buffers that are not ICE packets. Direct I/O allows data to be placed directly in the buffer of the calling user process. JBDRIVER will lock the pages containing the user buffers in physical memory and refer to the pages using Page Frame Numbers. JBDRIVER will use a buffered data path for large DMA transfers.

3.2.13.3 JBDRIVER outputs. The following shall be output from the JBDRIVER function:

a. ICE data to the buffer of the calling process.

b. Data blocks to the buffer of the calling process.

c. Status as a result of the appropriate Queue-I/O directive to the calling process.

d. Device characteristics data to the calling process.

e. Error data to the VMS error logging procedure.

3.2.14 Special requirements. TIS functions will require a Run Time Library that contains service routines, monitor routines, system table access routines, and routines to implement frequently called ICEHANDL functions. These routines will be reentrant and position independent.

The service routines will be buffer services, queue services and a logger routine. The buffer services routine shall provide allocate buffer and deallocate buffer functions. The buffers will be variable length with external chaining. The queue services routine shall provide an enqueue and dequeue function. The logger shall log PSS transactions to be appended to the print output for a specified job.

The monitor routines will be an error monitor and a history monitor. The error monitor shall log at a minimum I/O errors, system services error returns, internal logic errors, and depletion of TIS resources. The history monitor shall log user messages for users not logged in, major transactions such as starting a job, canceling a job, and file cleanup.

The ICEHANDL Status ICE and Error ICE routines shall build and issue Queue-I/O directives to the JBDRIVER to send TIS initiated ICE to the PSS. The system table access routines will provide a standardized, centralized method to ensure single-server access to critical TIS tables and other in-memory data structures.

3.2.14.1 Human performance. This section specifies the command interfaces between the TIS CPCI and the system users. The purpose of the specification of human performance is to define the major interactions between the software and the TIS users. To accomplish this interaction, the TIS CPCI shall provide the following three classes of commands:

a. Program development. The TIS CPCI shall provide interactive and batch commands to perform PSS job functions.

b. System control. The TIS CPCI shall provide interactive commands to control the software processing and to determine the status of the TIS.

c. Maintenance of PSS files. The TIS CPCI shall provide interactive and batch commands to add, display, delete, and change files in the PSS database.

The syntax and grammar of the TIS CPCI command language (TCL) will be modeled upon the SDSS VMS command language syntax and grammar, while incorporating similar command functions as the TACFIRE PDP-11 SPS, and providing enhanced user capabilities. The TIS CPCI command language will contain the following constructs:

a. An imperative verb that indicates which command is to be performed (for example, SHOW).

b. A noun that indicates the direct object of the verb (for example, SHOW STATUS).

c. A value that further qualifies the direct object of the verb (for example, SHOW STATUS JOBID, where JOBID is numeric).

d. A word that modifies the action of the verb (for example, SUBMIT/SCAN).

e. A word that modifies the direct object of the verb (for example, SHOW STATUS/FULL JOBID).

The TIS CPCI shall support two user functional roles: the TIS Manager and the PSS User. The TIS Manager shall have the capability to control the TIS software and queues, and to control the PSS processing. The PSS User shall have the capability to edit PSS source files, submit jobs, and determine the status of jobs. The TIS Manager shall be able to perform the PSS user functions in addition to the control functions reserved for the TIS Manager role. The PSS User shall not, unless enabled by the TIS Manager, be able to perform those functions of the TIS Manager that directly affect PSS processing.

The TIS CPCI shall provide the commands listed in table XVI. The TIS CPCI commands shall provide functions equivalent to all of the TACFIRE PDP-11 SPS commands, with the exception of the PQUE, RTRY, FORM, and EXIT commands, as described in the Design Description Document for Smart Peripheral System, Phase 3, Specification Number 595950-900, dated 20 June 1976. In addition, the TIS CPCI shall provide other commands to perform new functions. The TIS CPCI shall provide commands to support those functions previously implemented in the TACFIRE PDP-11 SPS Master User File Directory Maintenance Program (MUFD) CPCEI shown in Table XVII. The TIS CPCI commands are further specified in 3.2.14.1.1 and 3.2.14.1.2.

In the TACFIRE PDP-11 SPS, the FORM and PQUE commands are used in manipulating the SPS output print queue. Since in the TIS the print queue will be maintained and serviced only by the SDSS VMS, the FORM and PQUE functions will be performed through normal VMS commands as defined in the VAX/VMS System

Table XVI. Equivalence of SPS and TIS Commands

| Command from SPS Design Description 595950-900 | Equivalent TIS CPCI Command | Purpose or Function Performed |
|---|---|---|
| SEND | SEND | Transmits an operator message to the PSS. |
| ACTV | SUBMIT | Initiates a batch job to be executed by the PSS. The JCL file will be scanned for correctness prior to submission to PSS. The user may indicate that the print output is to be held for scrolling. |
| (None) | SUBMIT/SCAN | Checks PSS batch job control language before submission for execution by the PSS. |
| DACT | CANCEL | Cancels a previously submitted PSS batch job. The terminal user may optionally request a PSS dump. |
| (None) | CONVERT/VMS | Allows PSS block-formatted files to be edited by a VMS source editor. |
| (None) | CONVERT/PSS | Allows VAX-11 RMS format files created or edited by VMS to be processed by PSS in block-formatted mode. |
| IQUE | SET/QUEUE | Provides a maintenance capability for the PSS batch input queue. Allows operator control of input and output queues. |
| PQUE | (None) | This function shall be performed by the VMS software. |
| IDNO | SHOW STATUS/FULL | Allows detailed inspection of an entry in the PSS job identification and job number table; provides a detailed report on a specified PSS job. |

Table XVI. Equivalence of SPS and TIS Commands
(Continued)

| Command from SPS Design Description 595950-900 | Equivalent TIS CPCI Command | Purpose or Function Performed |
|---|---|---|
| AJOB | SHOW STATUS/ALL | Provides a report on all active batch jobs currently executing in the PSS. |
| STAT | SHOW STATUS | Provides a report on the status of the job entries in the PSS batch input queue for the terminal user. |
| (None) | SHOW Param | Allows inspection of the values of TIS CPCI variables that affect PSS processing and TIS CPCI outputs. |
| (None) | SHOW Perf | Provides a report on current internal statistics collected by the TIS CPCI that record processing loads. |
| IQUE | SHOW/QUEUE | Provides a report on the status of the input, print, and punch queues maintained by the TIS. |
| CLUP | CLUP | Stops execution of a PSS and performs recovery of queued batch jobs. |
| RTRY | (None) | This command will not be supported by the TIS CPCI. |

Table XVI. Equivalence of SPS and TIS Commands
(Continued)

| Command from SPS Design Description 595950-900 | Equivalent TIS CPCI Command | Purpose or Function Performed |
|---|---|---|
| STRT COLD | START/COLD | Initializes the TIS CPCI to provide processing for a PSS without input queue recovery. |
| STRT WARM | START/WARM | Initializes the TIS CPCI to provide processing for a PSS with input queue recovery. |
| FORM | (None) | This command will not be supported by the TIS CPCI. |
| BOOT | BOOT | Downline loads a PSS or other properly formatted load module through a SSIM. |
| (None) | TEST | Transmits a test message to a PSS through a SSIM. |
| (None) | SET Param | Allows changes to the values of TIS CPCI variables that affect TIS and PSS processing. |
| (None) | HELP Command | Provides a concise statement of the definition, syntax, and usage of a specified TIS command to the interactive user. |
| (None) | STOP | Causes the TIS software to exit and returns VMS to the status previous to execution of the TIS CPCI. |

Table XVII. Equivalence of MUFD and TIS Commands

| Command from MUFD CPCEI | Equivalent TIS CPCI Command | Purpose or Function Performed |
|---|---|---|
| CREATE | CREATE | Allows a PSS filename, eqivalent VMS directory specification, PSS filetype, and block size to be added to the PSS database. |
| DISPLAY | DISPLAY | Allows inspection of the contents of the TIS table of PSS filenames, equivalent VMS directory specifications, PSS filetypes, and block sizes. |
| PRINT | DISPLAY/PRINT | Provides the TIS DISPLAY function except that the output is printed on an SDSS lineprinter. |
| UPDATE | UPDATE | Allows entries in the TIS table of PSS filenames to be altered. |
| REMOVE | REMOVE | Removes an entry from the TIS table of PSS filenames. |
| EXIT | (None) | This command will not be supported by the TIS CPCI. |

Manager's Guide (AA-D027B-TE). These VMS commands will replace the FORM and PQUE commands used in the TACFIRE PDP-11 SPS.

The RTRY command directs the TACFIRE PDP-11 SPS software to retry unrecoverable PDP-11 SPS System Interface Unit (SIU), cardreader, or printer I/O errors. In the SDSS VMS, cardreader and printer errors (for example, line printer offline or cardreader pick check) will be handled through normal VMS procedures as specified in the VAX/VMS Operator's Guide (AA-D025B-TE). The SSIM will be tested or checked by the TIS TEST command instead of the RTRY command, and SSIM I/O errors will be retried up to four times by the JBDRIVER before an indication of a fatal hardware error is returned to the applications program.

The EXIT command in the TACFIRE PDP-11 SPS terminates the execution of the MUFD CPCEI. Since in the TIS the MUFD functions are not implemented as a separate program but have been incorporated into the TCL, the EXIT command is not necessary; instead the TCL mode will be terminated by a control-z sequence.

An example of the listing of the PSS on-disk catalog structure that will be provided by the TCL DISPLAY command is shown in figure 8.

The TIS CPCI should provide equivalent operator commands to the operator commands that are currently processed by the TACFIRE SPS to maintain similarity between the operation of the two support systems. For this reason, several of the command names from the TACFIRE SPS/PSS software have been retained in the TIS CPCI where the equivalent commands exist.

3.2.14.1.1 TCL user interface. The TIS CPCI shall provide two modes of interface to control, determine status, and submit input to the PSS:

a. Interactive.

b. Batch.

In the interactive mode, a VAX-11 RMS disk file containing PSS JCL can be created by using one of the standard VMS editors, or the JCL may be copied from another SDSS peripheral. The PSS JCL file may be submitted directly for execution by the PSS through the TIS SUBMIT command.

In the batch mode, a card deck containing PSS JCL can be submitted directly to the PSS for execution by the TIS SUBMIT command.

The execution of the TIS commands to submit jobs to the PSS, using VMS batch mode, shall not interfere with the facilities of the VMS cardreader spooler and VMS batch. If the SDSS VAX-11/780 cardreader is spooled and allocated to the VMS, then PSS job decks shall be read and executed by the TIS and PSS without SDSS VMS operator intervention to deallocate or de-spool the SDSS

TIS Version /X001/
PSS Master User File Directory
27-MAR-81 15:53:28

| PSS FILENAME | VMS EQUIVALENT DEVICE AND DIRECTORY | TYPE | DEFAULT BLOCK SIZE (FULL WORDS) |
|---|---|---|---|
| AUXFILE | DBA0:[AUXFILE] | 3. | 120. |
| BCMPSRC | DBA1:[BCMP.SOURCE] | 3. | 120. |
| BCMPOBJ | DBA1:[BCMP.OBJ] | 3. | 120. |
| BNAP | DBA1:[300,3] | 1. | 126. |
| DEVSRC | DBA1:[300,124] | 1. | 126. |
| DEVSRC80 | DBA1:[DEVSRC80.SOURCE] | 2. | 126. |
| ENVL | DBB1:[300,15] | 3. | 120. |
| FIELD | DBA3:[FIELD.V42.SOURCE] | 1. | 126. |
| FIELDOBJ | DBA3:[FIELD.V42.OBJECT] | 3. | 120. |
| FIELDSYS | DBA3:[FIELD.V42.IMAGES] | 3. | 2048. |
| NEAL02 | DBA2:[300,55] | 1. | 126. |
| PSOS | DBB3:[PSOS.OBJ] | 3. | 120. |
| PSS | DBA3:[PSS.IMAGES] | 3. | 2048. |
| PSSOBJ | DBA3:[PSS.OBJECT] | 3. | 120. |
| PSSSRC | DBA3:[PSS.SOURCE] | 1. | 126. |
| WSMRSPR | DBA2:[300,126] | 3. | 126. |
| WSMRSC80 | DBA2:[300,127] | 2. | 126. |

TYPE LEGEND: 1--SOURCE72, 2--SOURCE80, 3--OTHER.

Figure 8. Example of TIS Catalog Listing

cardreader. A TIS user shall have the capability to enter PSS JCL card decks into the TIS for execution (through the SDSS VAX-11/780 cardreader) and to employ both VMS and TIS commands in the same VMS batch stream.

No modifications or alterations to the SDSS VMS software shall be made in the implementation of the TIS PSS batch capability. The full capabilities of VMS batch shall not be altered by the TIS batch facility. One restriction (upon the capabilities of the TACFIRE PDP-11 SPS to be supported by the TIS) created by the VMS and the VMS cardreader driver (CRDRIVER) is that input PSS job decks cannot contain embedded punched card PSS object modules. These PSS object modules must be read in binary 12-bit code and the VMS cardreader spooler together with CRDRIVER do not suppport this capability for VMS batch input spooling of binary format data cards.

The TIS users shall have the capability to execute all of the TIS commands shown in table XVI and table XVII in the VMS interactive mode, that is, while logged-on to the SDSS VMS on a CRT or hardcopy terminal.

The TIS users shall have the capability to execute all of the TIS commands shown in table XVI and table XVII, with the exception of the TIS commands listed in table XVIII, in the VMS batch mode by submission of a VMS batch job containing TIS and PSS commands.

The TIS users (TIS Manager and PSS Users) shall execute TCL commands in TCL mode. TCL command mode shall be entered from the normal VMS Digital Command Language (DCL) mode by a single command; once TCL mode is entered, only TCL commands shall be recognized until the TCL mode is terminated by the TIS user.

A TIS user shall log into the SDSS VAX-11/780 by normal SDSS VMS procedures. The TIS CPCI will not require that a TIS user have extraordinary VMS privileges. The TIS user shall have the capability to execute TCL commands from any SDSS VMS account designated as authorized by the TIS Manager.

After the TIS user has logged into the SDSS VAX-11/780, the TIS user will be in DCL command mode, and may use any of the VMS facilities that the SDSS VMS account is authorized to use. The TIS CPCI shall not interfere with the execution of VMS commands while the TIS user is in DCL command mode. The TIS CPCI will not support the TCL command mode for compatibility command language (PDP-11 Monitor Console Routine (MCR)) mode SDSS VMS users.

The TIS user will enter TCL command mode by entering the TCL command in response to the VMS prompt. The TIS CMDINTRP process will then interact with the TIS user, and the TIS user will enter the appropriate functional role (TIS Manager or PSS User) and the TIS password that corresponds to the authorized TIS user. The password will not be printed on the terminal. Upon validation of the TIS username and password, the TIS user will be able to execute those

Table XVIII. TIS Commands not Supported in Batch Mode

| TIS CPI COMMAND | PURPOSE OR FUNCTION PERFORMED |
|---|---|
| CLUP | Stops execution of a PSS and performs recovery of queued batch jobs. |
| START/COLD | Initializes the TIS CPCI to provide processing for a PSS without input queue recovery. |
| START/WARM | Initialized the TIS CPCI to provide processing for a PSS with input queue recovery. |
| BOOT | Downline loads a PSS through an SSIM. |
| TEST | Transmits a test message to a PSS though an SSIM. |
| STOP | Causes the TIS software to exit. |

TCL commands authorized by the TIS Manager. While in TCL mode, the terminal user must not use the special VMS control-y and control-z command sequences to abort the TCL CMDINTRP process; instead, the TIS user will terminate TCL mode by entering a control-z sequence. When the TCL command mode is terminated, the TIS user shall be returned to the normal VMS DCL command mode.

The TCL command language shall prompt the TIS user for each parameter of a given authorized TCL command if the TIS user enters the TIS command imperative verb listed in table XVI or table XVII followed by a carriage return (CR) and no further TIS command parameters on the same line as the imperative verb. For example, if an authorized TIS user types the TIS SEND command followed immediately by CR, then the TCL will prompt for the system number, job slot number, and message text to be sent through the SSIM to the specified PSS job slot. A similiar prompting procedure shall be provided by the TIS CPCI for all of the TIS commands listed in table XVI and table XVII.

The TCL shall provide a help facility, similiar to the normal VMS help facility, while the TIS user is in the TCL command mode. The TCL help facility will provide a concise statement of the definition, syntax, usage, and parameters of a specified TIS command to the TIS user. While in DCL mode, the TCL HELP command shall not interfere with the VMS DCL mode help facility. The TCL help facility text shall not be added to the SDSS VMS help text libraries; instead, a separate TCL help text file will be created.

3.2.14.1.2 Functional role of TIS users. The TIS CPCI shall support two classes of TIS users:

a. The TIS Manager.

b. The PSS Users.

The TIS Manager shall have the capability to initialize the TIS, downline load a PSS into the MMP or an AN/GYK-12, to control TIS and PSS processing, and to maintain and alter the PSS on-disk database by any of the TIS commands listed in table XVI and table XVII. The TIS CPCI shall require that at least one TIS Manager be designated. The TIS Manager shall not be required to be logged-on to the SDSS VMS at all times; however, the TIS Manager shall be the only TIS user with the capability to execute all of the TIS commands. Messages to the TIS Manager originated by the TIS CPCI or the PSS shall be written on the Master SDSS VAX-11/780 VMS operator console (OPA0:) if the TIS Manager is not logged into the SDSS VMS.

The PSS Users shall have the capability to create PSS JCL files and PSS Tactical Procedure-Oriented Language (TACPOL) source files using the VMS editors, to submit jobs through the TIS to the PSS for execution, to cancel previously submitted PSS jobs, to convert PSS blocked-format source and JCL files to VMS format and to convert VMS format files to PSS blocked-format, to deter-

mine the status of the TIS and PSS, and to display the contents of the PSS on-disk master user file directory. The PSS Users shall not have the capability to execute the commands listed in table XIX, unless specifically authorized by the TIS Manager; otherwise, these TCL commands shall be reserved to the TIS Manager. Conversion or preprocessing utility functions shall not be necessary to edit a PSS source file or to submit a file of PSS JCL for execution; these conversion utilities are provided only to allow the TIS the capability to process any files from the TACFIRE PDP-11 SPS that might be stored in the obsolete PDP-11 SPS formats.

3.3 Adaptation. This section specifies, in descriptive and quantitative terms, the database requirements which affect the design of the TIS CPCI.

3.3.1 General environment. This section discusses the existing implementation of PSS disk file catalog structures in the TACFIRE PDP-11 SPS/PSS through the use of IAS/RSX Files-11 disk files. The TIS CPCI shall access the PSS disk file catalog structure defined in this section.

An IAS/RSX user identification code (UIC) or account consists of six octal digits in the format [group, programmer]. For example, the UIC [377, 374] has a group code of 377 and a programmer code of 374. The group and programmer codes must be in the range from 1 to 377 (octal) maximum. The UIC is used to partition the IAS/RSX files.

In the SPS the AN/GYK-12 resident operating system, PSS, uses disk files. In the TIS, all PSS disk files are stored on the SDSS VAX-11/780 disks using the IAS/RSX Files-11 file structures in Level 1 format or VMS Files-11 file structures in Level 2 format. A PSS file specifier is in the format:

filename,member,version

and an IAS/RSX files specifier is in the format:

Dev:[grp,prog]filename.type;version

Since the PSS file specifier does not contain the IAS/RSX UIC, a method is used in the TACFIRE PDP-11 SPS to convert the PSS file specifier to a IAS/RSX file specifier.

The PSS file name is made to correspond to an IAS/RSX UIC. A list of these correspondences is kept in an IAS/RSX disk file named

MUFD.DIR

The format of the PSS Master User File Directory (PSSMUFD) file that shall be processed by the TIS CPCI is shown in table XX. The PSSMUFD file shall allow the following entries:

Table XIX. TIS Commands not Supported for PSS Users

| TIS CPCI COMMAND | PURPOSE OR FUNCTION PERFORMED |
|---|---|
| CLUP | Stops execution of a PSS and performs recovery of queued batch jobs. |
| START/COLD | Initializes the TIS CPCI to provide processing for a PSS without input queue recovery. |
| START/WARM | Initializes the TIS CPCI to provide processing for a PSS with input queue recovery. |
| BOOT | Downline loads a PSS through an SSIM. |
| TEST | Transmits a test message to a PSS through an SSIM. |
| STOP | Causes the TIS software to exit. |

Table XX. Format of a Single PSS MUFD Entry

| FIELD NAME | SIZE (BYTES) | VALUE | DESCRIPTION |
|---|---|---|---|
| PSS filename | 8 | ASCII | Name of the PSS file corresponding to the VMS directory string. |
| VMS device | 4 | ASCII | Device name where the VMS directory string is found. |
| VMS directory string | Variable 1 to 31 | ASCII | VMS directory, subdirectory, or UIC that corresponds to the PSS filename. |
| Block size | 2 | Binary | Default block size for all members of the PSS filename. |
| File type | 1 | Binary | 1 — Source, 72 byte format.<br>2 — Source, 80 byte format.<br>3 — Miscellaneous (all non-source files) |

a. Up to a maximum of 8 ASCII characters for the PSS filename.

b. Up to a maximum of 4 ASCII characters for the VMS device where the corresponding PSS filename resides.

c. From 1 to 31 ASCII characters for the VMS directory specification. This directory specification may be either a UIC or an alphanumeric string, possibly containing subdirectories.

d. A value in the range of 1 to 2048 decimal for the default block size of all of the members of the PSS file.

e. A value in the range of 1 to 3 decimal for the PSS file organization type of all of the members of the PSS file.

An entry is deleted from the PSSMUFD file by zeroing every word of the appropriate entry within a particular block and then writing the block back to the file. A free entry contains all zeros.

Entries in the PSSMUFD for the directory specification may be VMS logical names.

Before a PSS filename can be used, an entry for it must be made into the PSSMUFD. The TACFIRE PDP-11 SPS MUFD CPCEI is a PSS catalog maintenance program, which can create new PSS filenames. The TIS CPCI shall provide functions equivalent to the MUFD CPCEI through the TCL user interface.

The following VAX-11 RMS file formats shall be processed by the TIS CPCI:

a. Sequential access, variable or fixed length records. This file organization is primarily intended for TACPOL-B source files and other PSS files containing textual data, which shall be directly editible, without conversion by the TIS users, by any of the VMS editors (for example, SOS, EDT, or EDI) and shall always be stored on the SDSS disks in this format.

b. Direct access, fixed length records. This file organization is primarily intended to support PSS data-oriented read and write functions, including shared access.

c. Block mode, virtual block number access with fixed length records equal to 512 decimal bytes. This file organization is primarily intended to support deblocking of PSS object and COMPOOL data files upon request by the PSS, including shared access.

The TIS CPCI shall support the use of the IAS/RSX UIC and the VMS Files-11 Level 2 directory specifier with the VMS maximum of 8 subdirectories.

The TIS CPCI shall access the PSS database on both Files-11 Level 1 media, for direct compatibility with the TACFIRE PDP-11 SPS, and on Files-11 Level 2 media, for operation solely in a VMS environment. The TIS CPCI shall access a combination of Files-11 Level 1 and Files-11 Level 2 files simultaneously, but a restriction of the VMS is that both Files-11 Level 1 and Files-11 Level 2 data cannot reside on the same physical disk pack or disk drive. The TIS CPCI shall allow access to TACFIRE PDP-11 SPS disk packs mounted as Files-11 Level 1 volumes.

One restriction of the TIS CPCI is that whenever new PSS filenames are to be added to the TIS PSS database, the appropriate information must be entered into the PSSMUFD file through the TCL commands specified in table XVII.

Another restriction of the PSSMUFD organization is that all files in the PSS database must have a VMS or IAS/RSX file type of DAT, except that VAX-11 RMS file specifiers for PSS JCL data may have a VMS or IAS/RSX file type of JCL.

When a PSS filename is specified for access, the TIS CPCI shall search the PSSMUFD file for the particular filename. If this PSS filename is not found, the TIS CPCI shall attempt to open the PSS file by using the specified PSS filename as the VMS directory string on the current user default device. If all necessary information, missing from the PSSMUFD, is present in the Open Command ICE Extension and if the PSS file is found in the subject VMS directory, then the TIS CPCI shall access the resultant PSS file. If these actions occur, the TIS CPCI shall inform the batch or terminal user by annotating the print output for the job to indicate the resultant file accessed.

3.3.2 System parameters. This section describes the data structures and constants required by one or more of the TIS processes that may change from time to time incrementally within a specified range according to operational needs.

The following data structures shall be maintained by the TIS CPCI:

a. The PSSMUFD, described in table XX, shall be maintained as a table in memory and shall be periodically paged out to disk.

b. The System Parameter Table (SPT) which contains the current settings and default values for TIS CPCI operating parameters.

c. The Job Input Queue (JIQ) which contains the entries submitted by interactive or batch commands for execution on a PSS.

d. The Job Control Table (JCT) which contains the current attributes of the jobs under execution by a PSS.

e. The File Attributes Table (FAT) which describes the characteristics of all files currently accessed by a PSS.

f. The Punch Queue (PUQ) which contains the entries currently in the punch queue.

g. I/O buffer pool. Variable length buffers are allocated and deallocated by buffer services routines in the TISRTL and VMSRTL.

h. The ICE Queue maintained in non-paged dynamic memory by the JBDRIVER.

i. The Tape Allocation Table (TAT) which describes the tape requirements of currently executing PSS jobs.

The TIS CPCI shall maintain the following ancillary on-disk files:

a. The Error Log.

b. The History Log.

c. The Message File.

d. The Help Text File.

e. JCL to be sent to the PSS upon receipt of the PSS Read Input Queue ICE.

f. Output to be printed for a specific PSS job.

The log, message, JCL, print output, and help text files shall be editible by VMS edit procedures, and shall be directly displayable to the TIS users.

3.3.3 System capacities. The TIS CPCI shall provide the following minimum capacities:

a. A maximum of 72 simultaneously open PSS files, extensible by a increment n set by the TIS Manager.

b. A maximum of 30 simultaneously executing PSS jobs.

c. A maximum of 25 entries in the JIQ.

d. A maximum of 25 entries in the PCT.

e. A maximum of 25 entries in the PUQ.

All other capacities will depend upon parametric settings by the TIS Manager.

# 4. QUALITY ASSURANCE PROVISIONS

The test and verification requirements, to ensure that TIS performs correctly and meets all of the requirements stated in Section 3, are specified in this section.

4.1 Introduction. The TIS CPCI shall be verified in accordance with the following requirements.

A TIS Test Plan shall be developed to ensure that all required functions specified in Section 3 are completely and correctly implemented. This Test plan shall cover unit level testing, system level testing, and acceptance testing.

4.2 Test requirements. This section states the requirements for each testing level.

The Unit Level Testing shall validate each major function specified in Section 3.2. Unit Level Testing will require a process to provide the inputs for the function being tested and that the outputs of the function are captured. Unit Level Testing will ensure that for each input the function produces the expected output.

The next level of testing, System Level Testing, shall validate that the TIS CPCI meets all the system level requirements stated in Section 3. This level of testing will require a test driver that simulates the PSS interface, as specified in Section 3.1. The test environment shall include the SSIM and a function generator to emulate the AN/GYK-12 IOX interface as discussed in 6.1.1. The System Level Test shall require a data capture function for test analysis.

The Acceptance Level Testing shall test the operational requirements of the TIS CPCI. An Acceptance Test Plan shall be written to establish the hardware requirements, the test configuration, personnel requirements, and test procedures to verify functional and performance requirements, inputs and expected test results. To ensure that SDSS operational requirements are met, the acceptance test shall include compilations and assemblies of TACFIRE modules. A system generation (SYSGEN) of the TACFIRE system shall be part of Acceptance Testing.

An Acceptance Test Report shall be written that covers all tests and inspections conducted in accordance with the approved Acceptance Test Plan. The report shall contain all test inputs, procedures, results, observations, and the hardware configuration. The report shall indicate the degree of compliance in meeting all system requirements.

Depending upon the TACFIRE SYSGEN configuration selected, DSC images of entire RP06 TACFIRE PDP-11 SPS/PSS database disk packs or selected portions of

the TACFIRE PDP-11 SPS/PSS database will be required for Acceptance Level testing.

5. PREPARATION FOR DELIVERY

5.1 VMS command files for installation of the TIS CPCI. The TIS CPCI shall include a set of VMS DCL command procedures to allow the compilation, linking, and installation of each computer program module comprising the TIS CPCI. These command procedures will contain sequences of commands frequently used during the maintenance of the TIS CPCI. Each command procedure shall include commentary explaining the purpose and processing that the procedure performs, in addition to the actual VMS commands. A written report shall be included with the delivery of the TIS CPCI that provides an overview of the maintenance and installation process.

6. NOTES

6.1 AN/GYK-12 I/O interface. The interface between the IOX or IOE and the SSIM consists of an electrical communications channel with the following major functional components:

a. Bidirectional information lines.

b. Data signals.

c. Address selection.

d. Device control.

e. Request, enable, command, indicator, and interlock lines.

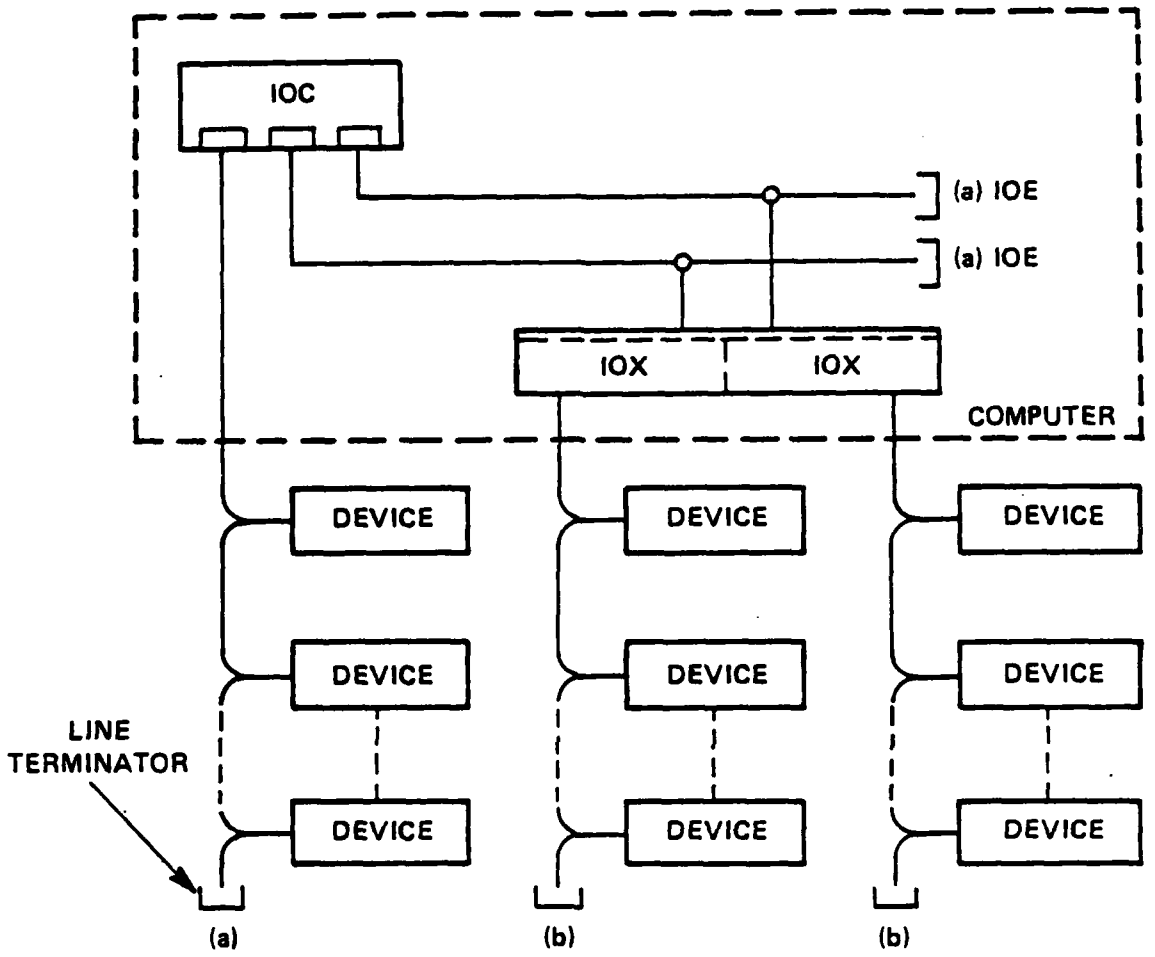
A simplified diagram of the IOX and IOE interfaces is shown in figure 9. The transfer of data between the CPU and peripheral device takes place over a cable containing 27 twisted pair lines plus a single line. The signals appearing on the lines are shown in figure 10. The set of the lines and the signals which appear on these lines are discussed in the following paragraphs.

Nine bidirectional information lines are used to transmit information between the computer and peripheral devices. The data transferred over these lines is in standard 8-bit byte plus parity format. The nine lines are used for the following purposes:

a. Data signals.

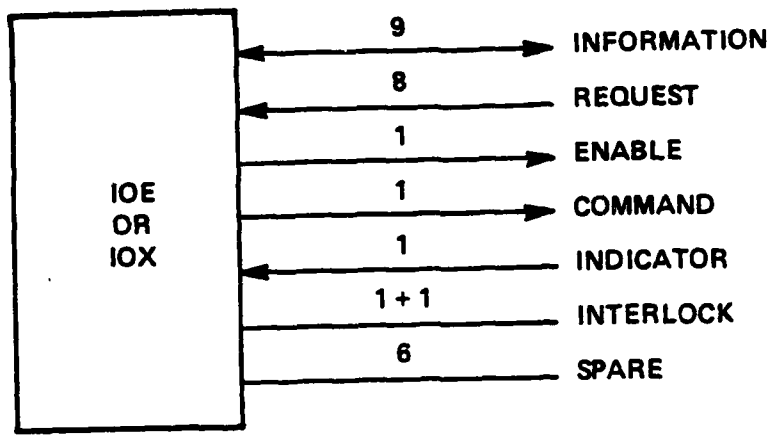
b. Address selection.

c. Device control.




NOTES: ALL CABLES SHOWN ARE 27 TWISTED PAIR PLUS SINGLE LINE FOR 55-PIN CONNECTION.
(a) DC INTERFACE. UP TO SIX IOES MAY BE CONNECTED TO THESE LINES.
(b) AC INTERFACE. UP TO EIGHT DEVICES MAY BE CONNECTED TO EACH IOX CHANNEL.

FIGURE 9. AN/GYK-12 INPUT-OUTPUT EXCHANGES




NOTE: Numbers Refer to the Numbers of Twisted Pair Except Interlock Which is One Twisted Pair Plus One Line.

FIGURE 10. AN/GYK-12 INPUT-OUTPUT EXCHANGE SIGNALS

Data signals are carried on the information lines. Information lines 0 through 7 contain the data byte, and information line P provides odd parity on the nine information lines during the data transmission phase of communication. When word transmission is used, four (4) bytes are transmitted sequentially, each with odd parity. The most significant byte is transmitted first in word transmissions (bits 0 through 7, followed by bits 8 through 15, bits 16 through 23, then bits 24 through 31).

Address selection is performed by the eight information lines (0 through 7) in conjunction with the Enable line or Command line to select a particular device. The device is selected if the address selection line corresponding to the device address switch setting is pulsed coincident with an Enable or Command signal.

Device control is performed by the eight information lines and the Command line. The information lines are used to signify specific operational actions to be performed by the device subsequent to an address selection phase occurring in conjunction with the Command line. The information appearing on the information lines specifies which of the operations is to be performed as shown in table XXI.

Each communication channel contains eight request lines with one request line assigned to each functional device connected on that channel. The request line utilized by a particular device corresponds to the device address selection switch setting on that device.

The Enable signal is used in conjunction with the Information lines to perform address selection. When this signal appears, the transfer of information that follows is data flowing to or from the computer or is a device interrupt. This signal is also used in conjunction with the Command lines to signify a Master Reset.

The Command signal is used in conjunction with the Information lines to perform address selection. When this signal appears, the transfer of information that follows is a command operation as shown in table XXI. Further information is predicated upon the actual command issued. This signal is also used in conjunction with the Enable line to signify Master Reset.

The Indicator signal is used to acknowledge receipt of a special command, and to initiate a device interrupt.

The interlock lines are used to indicate that the interconnecting cables are terminated. This line is not used by the device. The second interlock line is used to indicate that there are no disconnected connectors on the interconnecting cables. This line is short circuited inside the device. Neither line is connected to any circuitry within the device.

Table XXI. Device Control Formation by Use of the Information Lines

| SIGNALS PRESENT DURING CONTROL PHASE ON INFORMATION LINES | I/O FUNCTION |
|---|---|
| 0 and 3 | Device Command (DEV) |
| 0 and 4 | Output from Register (OFR) |
| 0 and 5 | Input to Register (ITR) |
| 0 and 6 | End of Block (EOB) |
| 0 and 7 | Device Stop |

Figure 11 illustrates the control sequences used in the AN/GYK-12 peripheral interface.

6.1.1 AN/GYK-12 modes of input-output operation. There are two classes of input-output operations in the AN/GYK-12 computer: programmed and automatic. Programmed communications are provided for Device Command (DEV), Output from Register (OFR), and Input to Register (ITR) instructions. Automatic operations are independent of instruction execution, take place at the peripheral device rate, and are controlled by I/O keywords and termination words.

Each addressed peripheral device responds to the following programmed I/O commands:

a. Device Command Operation (DEV).

b. Output from Register Operation (OFR).

c. Input to Register Operation (ITR).

The timing of these three instructions is independent of the peripheral device; however, an Indicator signal must be transmitted by the device to the CPU within 5 microseconds of the command (DEV and OFR) sequence to indicate the acceptance of command data (OFR) by the device, or that data has been transmitted (ITR) to the computer. If the Indicator signal is not received within the 5 microseconds, the device time-out bit of the AN/GYK-12 Indicator Register is set. This bit is used by the program to verify receipt of the command data.

The Device Command Operation (DEV) instruction is used to command the peripheral device to perform a peripheral-dependent function. The eight most significant bits are transmitted to the addressed peripheral device designated in the eight least significant bits of the DEV instruction operand and are interpreted by the peripheral to determine the operation to be performed.

The Output from Register Operation (OFR) instruction is used to set indicators or to provide a programmed output of a discrete word of four 8-bit bytes. The instruction causes the contents of the specified process register to be transmitted as four bytes to the address designated in the eight least significant bits (Device Address Byte) of the OFR instruction operand.

The Input To Register (ITR) instruction is used to interrogate a peripheral and to obtain a status message from a peripheral device or to provide a programmed input of a discrete word of one to four 8-bit bytes.

Automatic Input-Output operations are controlled by the status of two control words that are initially set by the computer program: the I/O Keyword and the I/O Termination word. The Keyword contains the mode of operation, the

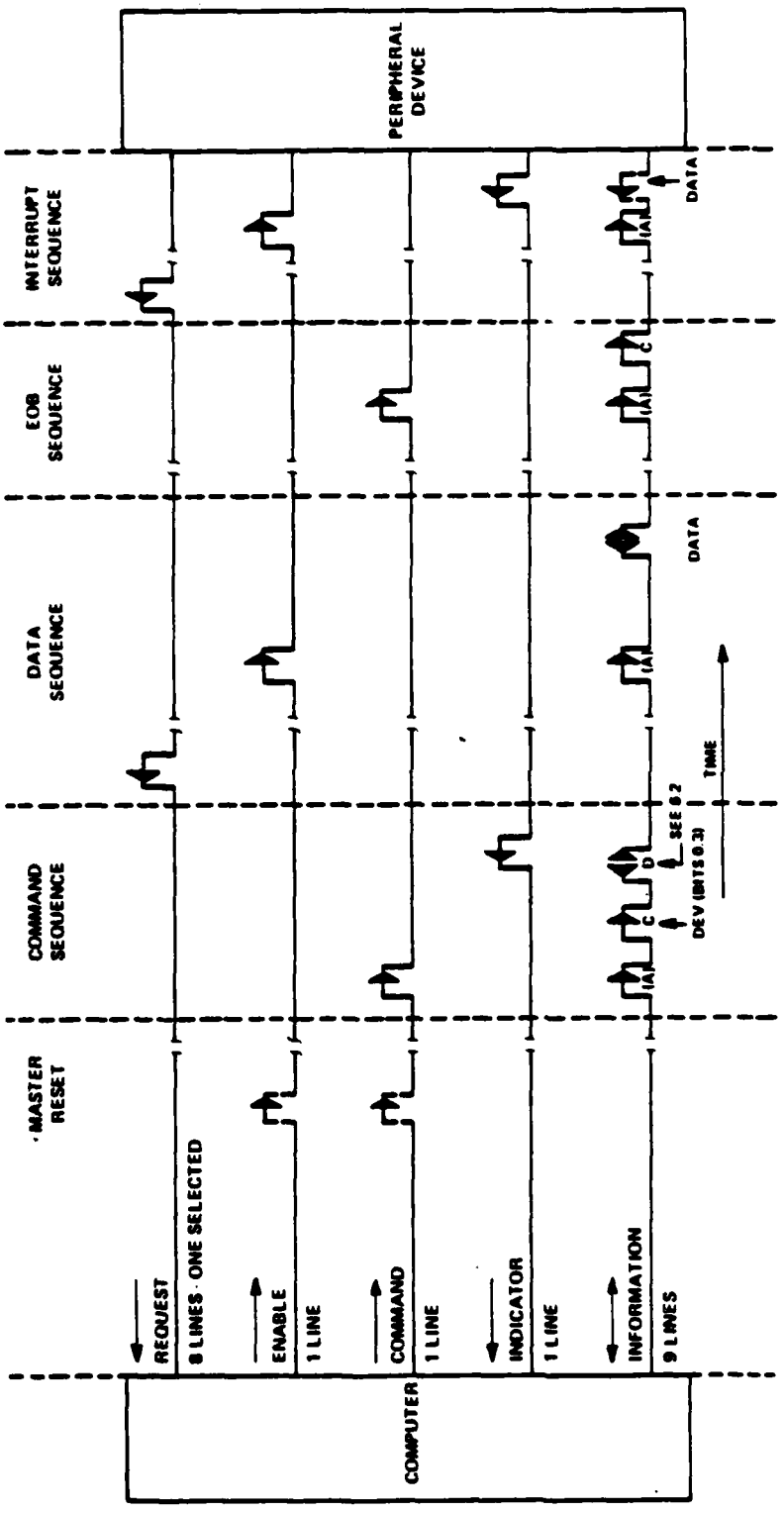



FIGURE 11. CONTROL SEQUENCES USED IN AN/GYK-12 PERIPHERAL CONTROL

block length, and the data address in memory, and is used during each data word or byte transmission by the IOU. The Termination word is used by the IOU at the end of a block of transferred data or upon fault detection for return of status to the software. The Termination word is also used to determine the program level to be entered when transmission is complete or when a fault is detected. There is a Keyword and a Termination word corresponding to each I/O device address.

The AN/GYK-12 performs four communication functions with the peripheral devices when in automatic I/O mode. The type of function to be performed is determined by the peripheral device by means of transmitted control signals. These function types are as follows:

a. Block of words (by byte) or byte transmission.

b. Alarm (clock) operation.

c. Device interrupt.

d. Bootstrap program load.

In block transmission operation, the interface buffer of the peripheral device is commanded to transmit or receive data by a programmed DEV instruction. The Keyword and Termination word associated with the device are initialized. The device then signals the CPU a request for service. Each time a request is acknowledged by the CPU, the proper Keyword is obtained from memory, a data transfer is made, and the Keyword is modified and returned to memory. When the Keyword block length field is decremented from one to zero, the Termination word is accessed, the finish-bit is set, and the computer sends an End of Block (EOB) sequence to the device. If the CPU detects a parity error during data transmission, the CPU accesses the Termination word and sets the transmission error bit; the data transfer operation is not terminated. If during the operation the device fails to respond to a command within 5 microseconds, a memory parity error is detected in a data or Keyword access, or a memory access violation occurs, the CPU accesses the Termination word, sets the operational error bit, and initiates the fatal-error program level (EPL). The block transmission is terminated and a stop command is transmitted to the device.

The alarm operation is similar to block transmission except that data is not transmitted and the current address field in the Keyword is not used.

The device interrupt sequence is initiated by a peripheral. After a device request has been acknowledged by the IOU, the device activates the Indicator signal. If the transmission error bit is clear, the IOU obtains the Termination word, sets the interrupt bit, and stimulates the program level specified by the normal program level (NPL) field of the Termination word. If

the transmission error bit is set, the EPL field of the Termination word specifies the program level to be stimulated.

The bootstrap load operation is initiated by a manual pushbutton on the IOU. When the IOU recognizes a bootstrap program load operation, a reset state is momentarily entered and a special device command is transmitted to the device with address $10_8$ or $11_8$. The IOU receives data from the loading device only during this operation. When the bootstrap program load operation is complete, the CPU enters program level $63_{10}$ and executes the instructions contained in the corresponding memory area.

Each peripheral device has a device address select switch which permits the selection of one of the eight addresses. The station address of the peripheral device is determined by the switch setting and the communication channel to which it is connected. The station address is a two-digit number; the more significant digit represents the communication channel numbers, and the less significant digit represents the device address switch setting. For example, when the switch in a peripheral device is set to position 3 and the device is connected to an IOX which is set to position 6, the station address will be $63_8$ (corresponding to device Keyword and Termination word for device $63_8$).

The station address with the lowest number selected has the highest priority for servicing. The operation of a peripheral device is the same, regardless of device connection to an IOX or directly to the IOU.

There are two types of address selection phases — one employing the Command line and the other employing the Enable line. The address selection phase, employing the Command line in conjunction with the Information line, precedes a device control phase. The address selection phase, employing the Enable line in conjunction with the Information line, precedes an automatic data transfer phase. The Information line (0 through 7), used in conjunction with the Enable line or Command line to select a particular device, corresponds to the setting of the peripheral I/O channel selection switch. The peripheral devices remain insensitive to data on the information lines until an address selection phase for that device occurs.

6.2 Timing tests of TIS CPCI design concepts. Timing results for mailbox I/O were obtained by creating a mailbox with a 64-byte record length and that contains one message (TIS mailboxes will contain more than one). The FILEOPS process puts the starting time into the 64-byte packet and writes the packet to the mailbox. The ICEHANDL process reads the packet, records the time of receipt, calculates the delta time, checks the packet contents, and begins the procedure again. Average timings were obtained as arithmetic means. Mailbox timings were run in two environments:

1. A loaded, busy system with four disk-bound users, three CPU-bound users, one user with constant terminal I/O, and two other relatively inactive users.

2. An unloaded, quiet system with no activity other than the test software.

Results are as follows:

1. Loaded system: Times of 10 msec to 625 msec to transfer one mailbox packet were observed. Based on a typical run of 1000 trials, an average time of about 150 msec was required to transfer one mailbox packet.

2. Unloaded system: Of any set of trials, approximately 80 percent of the I/O operations required less time than the resolution of the system clock, which is updated every 10 msec. Based on a typical run of 1000 trials, an average time of less than 10 msec was required to transfer one mailbox packet.

For global event flag timings, a named flag cluster was created by a timer program, and an event flag was set or cleared in the associated cluster. To set or clear an event flag, based on 1000 trials, an average of about 1 msec was typically required. This was measured in an unloaded system.

VAX-11 RMS I/O operations were timed by measuring the elapsed time required to write a given number of direct access, 512-byte records to an open file with an ascending record number, beginning at 1 and extending to a maximum of 1000. Times for open (depending on whether the file already existed) and close were also measured. Results are as follows:

1. Times of less than 10 msec to 110 msec to write one 512-byte record were observed. Based on a typical run of 1000 trials, an average time of about 10 msec was required to write one record.

2. If the target file did not already exist, the file open consistently required 150 msec.

3. If the target file already existed, the file open consistently required about 20 msec.

4. The file close consistently required about 120 msec.

These times were measured in an unloaded system. While this data is for an RK07 system, comparable times should be expected for an RP06 system.

Conclusions are as follows:

1. Mailboxes, global event flags, and global areas are of minimal implementation risk. The test software successfully incorporated all three of the proposed TIS interprocess communications techniques.

2. Although timing is certainly a function, in part, of system loading, the timings measured show that AST completion routines are not necessary for VAX-11 RMS FORTRAN I/O statements.

3. VMS Queue-I/O functions will be satisfactory to send and receive packets of information between processes using mailboxes. Two TISRTL common routines, GET_PACKET and SEND_PACKET, will be written to provide the same interface to the mailboxes for all processes.

4. Event flag clusters can be used to signal single events, for example, whether a given process has initialized successfully.

END

12-86

DTIC